DOCUMENT RESUME

ED 230 175 IR 010 695

AUTHOR	Zalewski, Donald L., Ed.
TITLE	Microcomputers for Teachers--With Application to Mathematics and Science. Topics for Teachers Series: Number 3.
INSTITUTION	School Science and Mathematics Association, Inc. Bowling Green, OH.
PUB DATE	82
NOTE	127p.
AVAILABLE FROM	School Science and Mathematics Association, 126 Life Science Building, Bowling Green State University, Bowling Green, OH 43043. ($6.00 per copy plus $1.00 postage. No postage charge on prepaid orders. $5.00 per copy for 10 or more).
PUB TYPE	Collected Works - General (020) -- Guides - Non-Classroom Use (055) -- Information Analyses (070)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Literacy; *Computer Managed Instruction; Elementary Secondary Education; Evaluation Criteria; Material Development; Mathematics Instruction; *Media Selection; *Microcomputers; Models; Programing Languages; Science Instruction; Teacher Education
IDENTIFIERS	*Computer Uses in Education

ABSTRACT

This collection of eight papers is designed to assist educators in becoming computer literate and in identifying and developing competencies necessary to implement microcomputers in K-12 classrooms. For the novice, it contains an overview of computers, a discussion of their use in daily life and in education, and coverage of important terms and features of microcomputer hardware and software. Examples are given of computer languages and of ready-to-use programs in science and mathematics. References are listed for individual chapters, and appendices contain sources of further information. For educators who must make decisions concerning hardware and software acquisitions, procedures are identified and explained. A systematic procedure for developing modules is detailed for use in creating programs. The papers include "Microcomputers and Teacher Education," by D. Zalewski; "Computers in Society and Education," by G. Bitter; "Microcomputer History and Developments," by J. Russell; "Microcomputers in Education," by W. Kraus; "Selecting Microcomputer Systems: A Process Model," by P. Barrette; "Computer Software," by H. Kepner; "Developing Materials for Microcomputers," by M. Agin and C. Simonsen; and "Programs from K-12 Science and Mathematics," by G. Marchionini and J. Camp. Appendices provide definitions of terms and clarifications, software information, and a software evaluation checklist. (LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED230175

SCHOOL SCIENCE AND MATHEMATICS ASSOCIATION

Topics for Teachers Series—Number 3

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

Minor changes have been made to improve reproduction quality.

Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

```
LIST

10  PRINT : PRINT
20  PRINT "MICROCOMPUTERS"
30  PRINT "FOR TEACHERS--"
40  PRINT "WITH APPLICATION TO"
50  PRINT "MATHEMATICS AND SCIENC
60    E"

]RUN

MICROCOMPUTERS
FOR TEACHERS--

WITH APPLICATION TO
MATHEMATICS AND SCIENCE

]
```

• MICROCOMPUTERS FOR TEACHERS— —

WITH APPLICATION TO MATHEMATICS AND SCIENCE

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

Darrell W. Fyffe

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR010696

Edited by Don Zalewski

SCHOOL SCIENCE AND MATHEMATICS ASSOCIATION

Topics for Teachers Series

Number 3

MICROCOMPUTERS FOR TEACHERS--
WITH APPLICATION TO MATHEMATICS AND SCIENCE

Edited by

Donald L. Zalewski

Northern Michigan University

School Science and Mathematics Association

Spring, 1982

Topics for Teachers Series

Number 1: CRYPTARITHMS AND OTHER
ARITHMETICAL PASTIMES (1979)
N. J. Kuenzi and Bob Prielipp (Eds.)

Number 2: INTERACTIONS OF SCIENCE
AND MATHEMATICS (1981)
by Peggy A. House

Copyright 1982

by the School Science and Mathematics Association, Inc.
126 Life Science Building
Bowling Green State University
Bowling Green, Ohio 43403

All rights reserved. No part of this monograph may be reproduced without written permission of SSMA.

FOREWORD

The School Science and Mathematics Association (SSMA) is pleased to publish this third volume in its Topics for Teachers series. Recognizing both the significance of the microcomputer in education and the critical role of the teacher, SSMA has produced this monograph to assist in the education of preservice and inservice teachers and other educators who will be involved with microcomputers in the classroom.

This monograph would not have been possible without the generosity of the editor, Donald Zalewski, and the authors who accepted invitations to contribute their time and expertise in writing the chapters. The SSMA is deeply grateful to them and to the members of the Publications Committee who assisted in the development of this monograph. In particular, we acknowledge the contribution of Michael Agin, Michigan Technological University; Arthur Coxford, University of Michigan; and Marilyn Suydam, The Ohio State University. We also thank Northern Michigan University for providing the word processing facilities and the manuscript typist, Patty Ogle.

We are pleased to present this monograph to teachers and teacher educators. We hope all who use it will find it a valuable resource.

Peggy A. House
University of Minnesota
Chairperson, SSMA
Publications Committee

Cover design by: Irene Tejada
School of Education
The University of Michigan

TABLE OF CONTENTS

	Page
FOREWORD	iii
LIST OF FIGURES AND TABLES	vii
PREFACE	ix
MICROCOMPUTERS AND TEACHER EDUCATION (D. Zalewski)	1
Computer Literacy for Teachers	1
Teacher Education and the Monograph	3
Conclusion	5
COMPUTERS IN SOCIETY AND EDUCATION (G. Bitter)	6
Computer Literacy	6
Society and Computers	8
Computers in Education	9
Educational Computer Uses	11
Implications of Computers in Education	14
Summary	18
MICROCOMPUTER HISTORY AND DEVELOPMENTS (J. Russell)	20
Background	20
Basic Computer Components	21
Educational Advantages of a Microcomputer	23
A Microcomputer Hardware System	29
Summary	32
MICROCOMPUTERS IN EDUCATION (W. Kraus)	33
Computer Languages	33
Computer-Assisted Instruction	36
Other Instructional Computing	40
Getting Microcomputers into the Classroom	42
SELECTING MICROCOMPUTER SYSTEMS: A PROCESS MODEL (P. Barrette)	46
Selecting a Microcomputer	46
Using the Process Model	47
Important Features to Consider	54
Conclusion	68

TABLE OF CONTENTS (continued)

	Page
COMPUTER SOFTWARE (H. Kepner)	69
Software	69
Finding Software	71
Educational Software Availability	74
Selection of Software	76
Developing Your Own Software	80
Conclusion	82
DEVELOPING MATERIALS FOR MICROCOMPUTERS (M. Agin and C. Simonsen)	84
Limitations of Available Programs	84
Constructing Modules: A Systematic Approach	86
An Application	90
PROGRAMS FROM K-12 SCIENCE AND MATHEMATICS (G. Marchionini and J. Camp)	95
Programs for Education	95
Examples of Applications Programs	96
Concluding Comments	109
APPENDIX A: TERMS AND CLARIFICATIONS	111
APPENDIX B: SOFTWARE INFORMATION	112
APPENDIX C: NCTM SOFTWARE EVALUATION CHECKLIST	114

LIST OF FIGURES AND TABLES

Figure		Page
3.1	Outline of a Computer System	23
4.1	A Simple Computer Program.	33
4.2	Machine Language Code with Explanation	35
5.1	Microcomputer Selection Process Model.	48
5.2	Sample Application Card.	50
7.1	Flowchart for Writing Chemical Equations	91

Table		Page
5.1	Microcomputer Features	55
7.1	Writing Chemical Equations	92

P R E F A C E

The computer age is upon us and schools are responding by incorporating computer education at all levels of the curriculum and by using microcomputers to facilitate teaching and learning. In order to prepare for and to implement these changes, teachers, supervisors, and administrators have started learning about the microcomputer and its uses in education. The School Science and Mathematics Association realized the importance and urgency of helping these educators become "computer literate" and produced Microcomputers for Teachers to assist in the task. Whether you are the learner or the teacher, the monograph can assist you in identifying and developing some of the competencies that are necessary to implement microcomputers in the classroom, particularly in the areas of science and mathematics for grades K-12.

For the novice, the monograph contains an overview of computers, a discussion of their use in daily life and in education, and a thorough coverage of important terms and features of microcomputer hardware and software. There are examples of computer languages and of ready to use programs in science and mathematics. The authors' reference lists and the appendices contain numerous sources of further information about microcomputers in education.

For educators who must make important hardware and soft-

ware acquisition decisions, well defined procedures are identified and explained. A systematic procedure for developing modules is detailed for those teachers who wish to create their own programs. However, you will need materials other than this monograph to learn a computer language, and hands-on experience is necessary to learn the actual operation of a microcomputer.

We hope that this publication is the beginning of many enjoyable experiences with the amazing microcomputer. The machine has great potential as a teaching/learning tool, and programming a computer to do one's commands can be an exciting adventure for you and your students. Good luck in your quest to become computer literate.

Donald L. Zalewski
Editor

Chapter I

MICROCOMPUTERS AND TEACHER EDUCATION

Donald L. Zalewski
Northern Michigan University
Marquette, Michigan

While the "computer age" began over three decades ago and sophisticated electronic technology has a great impact on our daily lives, computer uses and skills were only slowly being introduced into school programs since the 1960's. In the latter part of the 1970's, electronic miniaturization greatly increased public use and acceptance of computerization, resulting in pressures to accelerate incorporation of computer education into our schools. The development of the microcomputer made it economically posssible for school systems to begin responding to this demand. Although the responsibility of incorporating the microcomputer into school programs rests with the entire education community, the effectiveness of any efforts depends mainly upon one person--the computer literate classroom teacher.

Computer Literacy for Teachers

Numerous authors and groups who are concerned about computer education have offered various definitions of computer literacy and have identified a variety of backgrounds, goals and competencies that they feel a computer literate teacher should possess. (Moursund, 1980-81; Poirot et al, 1981; NCTM, 1980) Generally, it is felt that the informed teacher must realize the importance of computers in our daily lives

and in education, and must be able to use the microcomputer in the classroom. Specifically, the minimum list of knowledge and skills for the computer literate teacher should include the following:

1. The teacher should be able to discuss the importance of the computer, citing uses in the home, in business and in education.
2. The teacher must comprehend and use terms about microcomputer hardware and software.
3. The teacher should know the basic operational scheme of a computer.
4. The teacher must know the uses and limitations of microcomputers in the classroom.
5. The teacher must be able to operate microcomputer hardware in order to run prepared programs.
6. The teacher must find, select and use microcomputer software.
7. The teacher must be able to make informed choices of a microcomputer system's hardware.
8. The teacher should be able to read and write simple programs.

Every teacher, regardless of subject area, needs to develop capabilities such as those identified in 1-5 above. The size of the school system and the availability of a resident "expert" will determine how much expertise the teacher must have in competencies 6 and 7. However, the classroom teacher must be involved in making hardware and

software selections.

The eighth competency in the above list is debatable. Some teachers might think that they can make profitable use of the microcomputer by simply being able to run available programs. However, most computer education groups recommend that the teachers be able to read and write programs so that they can better understand and appreciate the potential of the microcomputer. Some groups also believe that teachers should learn enough about programming to develop their own software. Undoubtedly the programming skill that a teacher develops will depend on individual time and initiative.

Teacher Education and the Monograph

Although there is no agreement on a single list of competencies, there is agreement that we need informed teachers in the classroom. The main problem is how to help teachers become computer literate. For preservice teachers, the responsibility rests with well developed teacher education programs. For practicing teachers, supervisors and administrators, the colleges, professional organizations, and school systems must develop materials, conferences, seminars, workshops and inservice courses. The final responsibility rests upon the individual educator to attend available programs and to read informative materials.

This monograph can be used in several of the above learning situations to identify and to develop some of the competencies of a computer literate teacher. In Chapter 2,

Bitter discusses the important role of computers in our daily lives and the rationale, concerns and effects of using computers in education. He includes a comprehensive list of topics which could be used to develop computer awareness in teachers and students.

A brief background of computers and computer operations is presented by Russell in the beginning of Chapter 3. He identifies specific reasons and advantages which account for the microcomputer's popularity and concludes by identifying components of a complete microcomputer system. In Chapter 4, Kraus gives an overview of computer languages before identifying and analyzing the specific purposes and uses of microcomputers in education.

The difficult task of hardware selection is facilitated by Barrette's process model in Chapter 5. He includes a list of microcomputers and their features and discusses the key features of microcomputer system hardware. In Chapter 6, Kepner focuses on microcomputer software availability, sources and quality. He identifies criteria and explains a selection procedure for making intelligent decisions in software purchases.

In Chapter 7, Agin and Simonsen encourage teachers to develop their own microcomputer materials and they present a systematic approach for developing modules for classroom use. The final chapter contains examples of various types and levels of science and mathematics programs that were collected by Marchionini and Camp. References and appendices

provide additional terminology, sources for further study and a software evaluation checklist.

Conclusion

Microcomputers for Teachers is designed for educators who are responsible for including microcomputers in the K-12 mathematics and science program. It defines what is necessary to become computer literate and helps develop some of the knowledge and skills necessary to carefully and profitably implement the microcomputer in the classroom. Educators of all areas and levels can learn by reading and discussing the contents in either formal and informal settings. Teachers of mathematics and science will find examples for their classrooms.

References

Moursund, David. Introduction to Computers in Education for Elementary and Middle School Teachers. The Computing Teacher, Vol. 8, No. 3, 1980-81, pp. 6-11.

National Council of Teachers of Mathematics. An Agenda for Action Reston, VA.: NCTM, 1980.

Poirot, J., Taylor, R and Powell, J. Teacher Education. Topics: Computer Education for Elementary and Secondary Schools, New York: The Association for Computing Machinery, Inc., 1981, pp. 18-27.

Chapter 2

COMPUTERS IN SOCIETY AND EDUCATION

Gary G. Bitter
Arizona State University
Tempe, Arizona

In the last decade there has been a tremendous growth in the number of computers in existence. This has been brought about by the development of microcomputers so powerful yet so compact that they fit on a desktop. Microcomputers are playing important roles in businesses, classrooms and the home. With such an impact on society and in education, it is necessary that all people become computer literate. We will examine definitions of computer literacy and the effect of computer literacy on society and education. The implications and uses of computers in society and education also will be explored.

Computer Literacy

What is computer literacy? Ankers (1980) recommends that computer literacy include knowledge of hardware and programming. Others recommend that computer literacy should include explorations into applications and uses of computers whether it be in the office, classroom or home (Bitter, 1981; Bell, 1979; Carpenter et al., 1980). Jay (1981) defines computer literacy as consisting of three realities or components: an attitude, a knowledge base and the ability or capacity to perform some function or tasks with computers. Although definitions vary, computer literacy generally refers

to the knowledge of computers, their uses and limitations, how they work, and their impact upon society.

With increasing numbers of computers in existence, it becomes necessary to know what to expect from these machines and how to use them intelligently, productively and humanely. A positive attitude towards computers needs to be nurtured. Knowing how and where computers are used enables one not only to gain knowledge about computers but also to reduce the fear of them. We need to learn how computers are used in the home, business community and schools. Learning about new applications to aid society such as special uses in helping the handicapped helps us become computer literate. Finally, by using a computer we become familiar with it. A child might play a game on the computer, a secretary might write a business letter on it, or a scientist might obtain statistical data by using the computer. Some people might even learn the art and science of computer programming. All of these reflect some degree of computer literacy in relation to the role of computers in society and education.

Why is it so important to become computer literate? "Homeowners hiding from the computer revolution are only delaying the inevitable," says the Arizona Republic (1981). It continues by emphasizing that by the end of the 1980's eight to ten million homes will have computers and up to 45 million will have access to information via computer terminals in the home. There were only 10,000 home computers in 1977. In 1981 there were already more than one million

computers (micro, mini and main frame) in use.

A survey indicated that 400,000 microcomputer units existed in 1980 with 46,000 being used in schools They estimated that there will be about 600,000 units with 105,000 to be used in schools by the end of 1982. Jay (1981) speculates that there will be 40 million computers by 1990, and 39 million people will have to know about them and how to use them. These numbers indicate that computer literacy will be necessary in the 80's!

Society and Computers

Computer literacy can be just as important as being able to read or do mathematical problems. In terms of daily operations, it is not necessary for individuals to become experts in using electricity and electronics to operate a television set. Likewise, it is not required to understand how a computer works or to be familiar with the components of a computer such as a microprocessor or memory. In other words, operating a computer is much like running an appliance. Of course, to get a computer to read, write and compute involves more training; you must learn a computer language to communicate with the computer. This skill adds to the fun for some, provides problem solving capabilities or a hobby for others, and can provide career opportunities.

Computers can be used for solving complicated problems as well as for playing games for fun and relaxation. Many of the arcade games children and adults play are available on home computers. Instead of passively watching television, the

children and adults can be actively involved in a pursuit. What's more, by playing games, children and adults become less anxious about computers. This is an important goal of computer literacy.

Home computers can help solve many family and business bookkeeping problems. Many computer programs already exist to help with personal income taxes, shopping and budgeting. With the addition of special telephone devices it is possible to gain access to information of all types. Stock market reports are available and the computer can be used for electronic mail. The list grows daily with new applications coming on line in many industrialized countries.

Computers in Education

"Computers in education" can be a rather nebulous phrase simply because it covers so many ways that the concept can be applied. The phrase usually includes the formal applications that enable teachers and students to reach educational objectives. Microcomputers may be used for educational pursuits in the home by parents and children. In addition, there is the learning that goes on about the computers themselves (Mazur, 1980).

Positive Factors

The positive factors presently influencing schools toward microcomputer involvement are low cost, individualized applications and curriculum options (Bitter, 1980). Many schools are purchasing microcomputers on an experimental basis. They

feel that it is important not to be left out of this revolution. Teachers look upon the microcomputer as an extra helping hand and as a unique method of teaching low and average, as well as high ability students. Many educators are also interested in the classroom management options which are possible with a microcomputer.

Concerns

With the growing acceptance and use of computers in education, several concerns have arisen. First, there is a general anxiety toward technology and technology assisted learning (Bitter, 1981). Feelings of fear and tension pervade the first-time user. Hardware and software are concerns. There is worry that the hardware will not consistently work without costly repairs. Software availability for specific hardware is also a justifiable concern. A major problem is that software is not interchangeable from one machine to another.

Boredom in computer assisted lessons can be a common problem unless good software is used. Teachers fear that initial interest will fade with increased involvement, bringing about a decline in learning. Teacher preparation in the use of the microcomputer and the ability to apply it effectively to curriculum needs is also a valid concern. Teachers need to be adequately prepared to understand the microcomputer and its implications in the classroom. Of course there are many other issues which will be raised in the years ahead.

Educational Computer Uses

There are three aspects of computer education of significance to educators. They are computer awareness, computer managed instruction and computer assisted instruction. Each one is introduced in this chapter and discussed more thoroughly in later chapters.

Computer awareness was summarized earlier as knowing about a computer and using one reasonably well without having to become an expert. Basically, the situation should not exist where a person is afraid to walk up to a computer and type in some information. The following outline suggests some computer related topics which might be included in the curriculum to develop computer awareness.

I. Computer Capabilities

- A. What a computer is
- B. Science fiction discussion
- C. Actual capabilities (speed, accuracy and repetitiveness)
- D. Programming
- E. Flowcharting

II. History of the Computer

- A. Abacus - 3000 B.C.
- B. Pascal - first mechanical adding machine - 1642
- C. Babbage - mechanical computer - 1830
- D. Hollerith - developed punched cards - U.S. census - 1900
- E. Analog computer - 1931
- F. Mark I - first general purpose computer - 1944
- G. ENIAC - 1945
- H. Werner Von Braun - computer technology
- I. IBM - 1959

III. Computer Generations

- A. Vacuum tube (first) - 1946
- B. Transistor (second) - 1959

C. Integrated circuit (third) - 1964
D. Large scale integration (fourth) - 1978
E. Very large scale integration (fifth)

IV. How a Computer Works

A. Accessing a computer
B. Input and output types
C. Arithmetic unit
D. Central processing unit (CPU)
E. Memory
    a. Types (disk, magnetic tape, magnetic core)
    b. Read only (ROM) and random access (RAM)
F. Software - Hardware
    a. Languages
    b. Stored program
    c. Parts of a computer
G. Compare computer operation to human solving

V. Computer Types

A. Computer Systems
B. Mini-computer
C. Micro-computer
D. Time sharing computers
E. Microprocessors
    a. Appliances
    b. Automobiles
    c. Robots

VI. Home Computer

A. Types
B. Applications
    a. Passive
    b. Active

VII. Computer Uses

A. Society
    a. Banking
    b. Department stores
    c. Medicine
    d. Law enforcement
    e. Government
    f. Weather prediction
B. Education
    a. Computer assisted instruction
    b. Computer managed instruction
    c. Computer extended instruction
C. Data Processing

VIII. Computer Misuses

A. Fraud
B. Identification codes, pass words and data files
C. Privacy (credit checks)

IX. Computer Careers

A Computer operator
B. Keypunch or coding
C Programmer
D. Systems analyst
E. Computer scientist

X. Future and the Computer

A Individual Predictions
B. Special Topics
   a. Speech synthesis
   b. Robotics
   c. Automobiles
   d. Home conveniences
   e. National Data Banking
   f. Medicine
   g. Organ implants
   h. Space
C. Microelectronic Industry Predictions
D. Cost
E. Artificial intelligence

Computer managed instruction is the use of the computer to keep data on the student and to make curricular and instructional decisions. It includes programs for student attendance, scheduling, records and progress. Grade book record keeping is also available. Many instructional programs include built-in management systems. Several major publishers now market sophisticated computer managed instruction systems for education.

Computer assisted instruction (CAI) is the use of the computer to actually simulate teaching and learning situations. It includes drill and practice, tutorials, simula-

tions, games and information retrieval. These important functions are detailed in Chapter 4.

One additional CAI application is problem solving, an area so vital that it was prominently mentioned by the National Council of Teachers of Mathematics in its "Agenda for Action." Their first recommendation was that "Problem solving be the focus of school mathematics in the 1980's" (NCTM, 1980). Problem solving with the microcomputer involves the ability to program the machine to solve problems. Therefore, it will be necessary for some educators to learn the microcomputer's language (i.e. BASIC) in order to take full advantage of its use for CAI.

Implications of Computers in Education

There are many unanswered questions related to the successful introduction of computers in education. Little research involving the microcomputer and children is available. The discussion which follows is a general summary of the potential and possible implications for utilizing the microcomputer in education. Computers can help educators perform many tasks. At the same time they should not be considered a panacea or substitute for good teaching! The microcomputer is only a new tool to assist the teacher and one must take into account the goals which one is trying to accomplish when using it.

Computers can provide a teacher with the option of presenting a subject in different ways appropriate to children with different styles of learning. Therefore, a computer is

a tool that can assist the teacher and the child in the learning process. However, it must be used wisely to avoid harmful effects; using the computer to the exclusion of other methods can limit a child's experiences. A program which requires thinking and problem solving can do much for the child. But if the program is written so that a child can "con" the answer from the computer, then obviously the child does not benefit. We know many children turn off to learning simply because they do not get any feedback or other responses from other teaching methods. For these children, a well conceived CAI program can provide motivation that might not otherwise be available. (Olds, 1981).

Microcomputers should be considered as a valuable aid in all instruction. For example, computers can be used to teach music concepts. Any teacher who teaches music theory or anyone who has taken music theory knows this can be very challenging. Yet the computer has proven to be "possibly the most flexible of all instruments." (Jones, 1980). The computer can go over concepts repeatedly without losing patience as many instructors do after an extended period of time. Furthermore, students can experiment with the computer and explore numerous questions. Finally, concepts that are difficult and time consuming to present can be handled more effectively with appropriate microcomputer software, especially in the area of graphic displays.

Many curriculums follow some general sequencing format. Mathematics is usually taught hierarchically; one must learn

to add single digit numbers before one can add and carry in multi-digit addition problems. Therefore, many different types of presentations will be needed to satisfy individual learning paces and content levels. Because computer programs have the capability of branching, they can provide the pacing, the teaching approaches and the remediation as necessary. This implies not only that programs must be well written using sound educational principles and strategies, but also that the programs must be diversified enough to benefit a relatively large number of children.

Primary curriculum areas including mathematics, reading and spelling frequently utilize computer assisted drill and practice programs. The function of the programs is to drill students in practice sessions while at the same time determining if the child has achieved the desired level of skill for a particular task. The implication is that as microcomputers take over this particular task, teachers will be free to use that extra time to give special attention to students who need it. Educators no longer will have to teach to the greatest number in hopes of reaching the most students. This is because microcomputers are capable of allowing children individually to master the particular skills at their own pace.

Progress reports can be made a part of a computer program. This would enable the teacher to keep track of the student's progress without the usual drudgery and time lapse. Having access at all times to a student's progress, a teacher

can make adjustments in a child's educational objectives to help the child reach goals continuously during the learning process. Perhaps with this information, children who are highly vulnerable to failure can receive attention and help early and stand a better chance of success.

In the future, curriculum materials can be designed so that a child can take the homework home on a disk or tape, complete the assignment on a home computer, and return the disk or cassette the next day. The progress report will be recorded on the tape or disk and can be reviewed by the teacher (Mazur, 1980). The obvious assumption is that all computers are compatible. Unfortunately, they are not so today. As technology advances and standardization becomes necessary, then this idea may be feasible.

No matter how the microcomputer is used in the classroom, it is mainly a helping hand for the teacher. It is hoped that teachers will have more time available to do other tasks and with this time the teacher can concentrate on the job for which he or she was hired--to teach.

Summary

Computer literacy is becoming extremely important in education and society, especially since the microcomputer age has exploded around us. It is estimated that by 1990 there will be over 40 million computers in use in the United States. With only one million in use today, it is obvious that we will need to actively involve our students in order for them to become computer literate.

The inclusion of computer literacy in the curriculum will depend upon the type of programs written, the quality of the programs, and the availability of quality programs that meet the needs of the schools. Future developments in graphics, sound or speech synthesis capabilities, home information systems and voice recognition will provide the continuing potential of improving microcomputer applications in the classroom as well as in the society. However, no matter how sophisticated the computer or programs become, they are only a tool and will not replace a good teacher. Remember:

> A Computer has no conscience
>
> It feels no passion
>
> It has no life of its own nor any sense of value
>
> It is neither a poet nor philosopher
>
> A computer in short is a machine, not a fool or tyrant.
>
> Source unknown

References

Ankers, Joan, et al. Classrooms Make Friends With Computers. Instructor, February 1980, pp. 52-58.

Barstow, Daniel. Computers and Education: Some Questions of Values. Creative Computing, February 1979, pp. 116-119.

Bitter, Gary G. Computers and Computer Resources. Teacher, 97 (5): pp. 62-63, 1980.

"Computer Literacy Needs." Microcomputers in Education, Arizona State University. Tempe, AZ, 1981, pp. 1-10.

Exploring With Computers, Julian Messner: New York, 1981.

Microcomputer Applications for Calculus, Prindle, Weber and Schmidt: Boston, 1972, Revision, 1982.

Bitter, Gary G., et al. McGraw Hill Mathematics Series, Grades K-8, 1981

Carpenter, Thomas, et al. The Current Status of Computer Literacy. The Mathematics Teacher, December 1990, pp. 669-673.

Jay, Dr. Timothy B. Are You a Computer Literate? The Computing Teacher, Volume 8, No. 4, Academic Year 1980-1981, pp. 58-59.

Jones, Beverly. Computers in the Arts and Humanities. The Computing Teacher, Volume 8, No. 6, Academic Year 1980-81, pp. 22-29.

Mazur, Ken. "Grow Old Along With Me! The Best is Yet to Be..." Creative Computing, August, 1980, pp. 39-42.

Molnar, Andrew. "The Coming of Computer Literacy: Are We Prepared for It?" Educational Technology, January 1981, pp. 26-28.

National Council of Teachers of Mathematics, An Agenda for Action." Reston, VA, 1980.

Olds, Henry F. Jr. The Making of Software. Classroom Computer News, July-August, 1981, pp. 20-21.

"Show Plugs Home Computers." Arizona Republic, July 25, 1981, p. 3.

Chapter 3

MICROCOMPUTER HISTORY AND DEVELOPMENTS

John C. Russell
University of Wyoming
Laramie, Wyoming

Background

Simply put, a computer is an electronic device which is manufactured to accept an ordered sequence of instructions given to it in an appropriate language and to carry out those instructions with great speed and accuracy. The sequence of instructions is called a "program" and we say a computer is "programmed" when it has received and can execute those instructions.

Some say Charles Babbage was responsible for the development of the computer in the early 1800's when he conceived his "difference engine" and his "analytical engine." Others suggest Harvard's Howard Aiken started things between 1937 and 1944 when he developed the Mark I, and still others would credit John Von Neumann because of his notion of a machine under the control of a stored program. Regardless of where the credit is given, the computer is with us and is being rapidly refined. Having passed through eras of vacuum tubes and of transistors, we are now in a generation in which the highly miniaturized integrated circuit printed on a silicon chip has become the keystone. Today, chips that cover a fingertip contain thousands of paths that allow electrical currents to perform hundreds of mathematical operations literally in the blink of an eye.

Computers come in a variety of sizes and capabilities. Large versions called mainframes or "maxis" might be found in the Pentagon helping preserve our national security. Middle-sized computers are called minicomputers or "minis" and would be appropriate to keep personnel records and do the payroll for a small company. But it is neither the maxi nor the mini which is sending waves of change, excitement and concern through the educational community. The hottest item on the school scene is the fledgling microcomputer, or "micro" for short. A micro works in principle the same as its bigger cousins, but it features a versatility and modest price tag that make it very attractive to schools.

Basic Computer Components

For the educator trying to become familiar with computer technology, the vocabulary can be overwhelming. Two frequently used terms are "hardware" and "software." Hardware describes the intricate circuitry and encasements that were stamped out at the factory, while software is comprised of the prerecorded programs that breathe life into the otherwise inert hardware and of the documentation that describes the operation of the computer and the running of the programs. The more likely it is to break when you drop it, the more likely it is hardware.

The single most important piece of hardware for any microcomputer is the "microprocessor." This tiny wafer is the vital component which controls the flow of signals throughout the computer system, performs arithmetic calcula-

tions, makes decisions and tells any peripheral devices that might be connected when to perform and when to cease activity. In the larger computers, these functions are handled by the central processing unit (CPU).

Two other components of a microcomputer system are "input" and "output" devices. As fancy as a microprocessor might be and as rapidly and efficiently as it might thrust commands and data on their way through the labyrinth of circuits, it is of no avail to a human user if one cannot direct things from the outside or receive understandable results of the computer's labors. Typical input devices such as keyboards, card readers and light pens permit the user to enter instructions and data into the computer, while a TV monitor or line printer are examples of output media. In large computer systems, much input and output is produced at remote locations through a component called a "terminal."

A microprocessor deals with instructions and data, often a great number of each. Consequently, an important part of any computer system is "primary memory." In most micros in schoolrooms today, primary memory comes in two forms: read-only memory (ROM) and random access memory (RAM). ROM usually stores instructions which are of permanent importance to the computer system and are recorded at the time of the micro's manufacture. Those instructions are impossible for the user to alter. RAM, in contrast, is a memory area in which a user stores his program of the moment, and its contents can be altered or erased by a variety of simple means.

The final major component of a computer system is "secondary storage." When user programs are important enough to be worth saving or when the data generated by a program will be put to future use, these programs and data can be preserved by recording them on a medium such as a magnetic tape or disk.

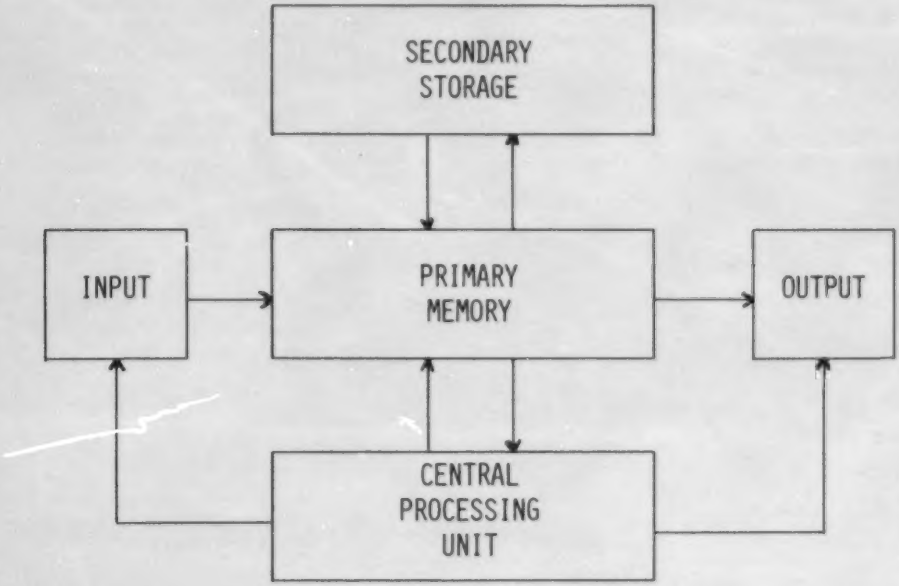


Figure 3.1. Outline of a Computer System

Educational Advantages of a Microcomputer

The computer's presence was first felt in precollege education when terminals were placed in the school--most likely in mathematics or science classrooms. More often than not, the host computer was at a distant campus or business and communicated with the terminals only through a telephone line. Some schools purchased minicomputers, but these often

were used mainly for administrative purposes.

Terminals and minicomputers are still used in schools around the United States. But microcomputers have launched a full-scale invasion of the territory held by the bigger computers. What has been the reason for the overwhelming interest in the micro? What would make large city and state school systems adopt a certain micro system as its own? How can such little computers as Apple, TRS-80, Pet and Atari win so many new friends when everybody knows IBM, Control Data, Digital Equipment and other big companies have been the computer authorities for a much longer time? There are numerous reasons for the microcomputers' popularity and they include important differences between micros and larger computers. Here are a few important considerations:

1. Expense. Perhaps the most important reason for the microcomputer's popularity is its low cost. The purchase price of a basic microcomputer is as low as $500 and with necessary add-ons will generally double that figure. A basic system can be bought for only about $1,500. Since most school systems which need access to a computer cannot afford the six- or seven-figure prices of a mainframe, they usually opt for use of terminals or minicomputers. But when rentals, telephone costs and the disadvantages of these alternatives are taken into account, micros often provide a cheaper investment.

2. The literacy issue. In the 1980's, educators are recognizing and promoting the importance of creating a computer literate public. In developing computer literacy, it

makes little difference whether the available hardware is a micro or a terminal to a larger computer. But in one important way the micro is more useful to have around: the entire system sits within the student's view. The input, output and storage devices are obvious and one must only take the cover off the microcomputer to see the microprocessor and memory units.

3. Future contact. Computers are predicted to be in every home in the relatively near future. The low cost and future developments will make a microcomputer in the home as common as a television set or a hand-held calculator. Is it not true, then, that schools should put in the forefront the kind of equipment the daily computer consumer will want to be most familiar with?

4. Portability. When computers were introduced into schools, they were almost universally found in the high school mathematics department. But programs for micros have something for everyone whether it be in social studies, mathematics, English, art, music or physical education. Serving this versatility is the fact that micros are portable, being no larger or heavier than a desk typewriter. They can be put on a cart and whisked down the corridor to reconnect with the learning process in another department with little trouble. Larger computers and terminals are bulkier and heavier, and they often require special environments.

5. Simplicity of use. To work on a large computer requires dealing with a system involved in handling many

users at a given time (called time-sharing), keeping track of countless files and, in short, supporting a complicated hierarchy of operations and responsibilities. As fascinating as this indeed is, the price the user pays is having to learn a more complex set of commands than to work with a micro. The simpler the process of getting to talk to a computer, the more quickly will computer anxiety be dissipated.

6. Color graphics. The major brands of microcomputers all have a variety of vivid colors and the ability to animate lessons and games to keep the attention of the learners. A commercial color television set is all that is needed to display those effects. The larger computer systems that are available to school districts generally do not have these capabilities.

7. Sound. Like color and animation, sound can be used to highlight and reinforce the learning experience. From simple squeaks and bells to intricate melodies in more than one voice, microcomputer sound systems extend the potential of the lesson or the enjoyment of competition as no terminal will do. Some music classes are incorporating micro systems in theory and composition to the delight of instructor and pupil alike.

8. Interfacing. The music teacher may wish the computer to drive a music synthesizer; the administrator may wish it to read daily attendance cards; or the special education teacher may want a certain student to be able to maneuver a flashing dot ("cursor") on the screen as opposed to

typing responses at the keyboard. The process of connecting special devices to the computer is called "interfacing" and here again the micro shines. The manufacturers have allowed for interfacing by providing ports or slots where the user can hook up a device purchased right off the dealer's shelf. This flexibility is not so readily available with a mainframe terminal.

9. Networking. With several micros in a school, it is probably most common to see individual students doing their own programming or individual teachers in their separate classrooms running simulations or giving demonstrations. But one variation is the networking of several micros into one system. Given a network, a teacher may choose his micro to be the "boss," thereby controlling the programs that are run on the student micros around the room and providing an organization that the teacher feels more comfortable with.

10. System reliability. All machines break down from time to time. Computers are no exception. Big ones and little ones have components that fail. All terminal users are out of business when their mainframe goes down and they remain out of business until the computer is repaired. However, the school that chose to buy several micros is only inconvenienced when one of its machines develops a problem. While the ailing machine is being repaired, the remaining micros still function, sometimes even taking up the slack.

11. Software availability. All of the previous considerations are important ones in the selection of an appro-

priate system, but there is not a more crucial issue than that of software availability. Without the programs to give it commands, no computer, regardless of size, will be able to run drills, tutorials, simulations or perform classroom management and administrative tasks. It is clear that a school computer consumer must be very sure that software support is available before spending money on hardware.

Presently, there is a variety of educational software available for both the mainframe and microcomputers, but the selection of high quality programs is limited. Because micros seem to be the direction in which schools are going, publishers interested in the education market have put more efforts and resources into creating software for the small machines.

One might observe a lack of microcomputer programs in a given area and suggest that the teachers can learn a little programming and do the writing themselves. Make no mistake here: the creation of educationally valid, technically bug-free software is a demanding and time-consuming activity! A teacher cannot teach all day, correct papers and plan in the evening, and have time to write major computer programs. A school system must recognize this and the consequent need to budget released time or else it should plan to find software elsewhere.

One further point should be made here. Do not erroneously conclude from their youth that micros are toys capable only of playing those interminable space and fantasy games

which unfortunately receive so much publicity. Micros are the product of our latest technological developments and their potential educational applications are limited only by the creativity of software developers and the informed classroom teacher.

A Microcomputer Hardware System

As amazing an instrument as the micro is, it is impossible to show its stuff without being connected to one or more of a variety of devices referred to as "hardware components." Just as music buffs commonly put together a stereo component "system," possibly acquiring amplifier, turntable and speakers from more than one source, micro users coordinate a system useful to them and their particular needs.

First, our consideration logically turns to entering information into our micro. The most common input device is the typewriter-like keyboard. This comes as standard equipment with microcomputers. If there is a difficulty with a keyboard entry system, it is that only one person can use it at a time, and with a hunt-and-peck typist in control that time can be considerable. One way around that is to interface a card reader to the machine. Cards the size of the often maligned "IBM card" can be coded in pencil and quickly run through the reader with the same effect as if the information had been keyed in. Additional input devices will be discussed later because of their joint input and output functions.

The most common output device is a television-like screen

which is either a monitor, a CRT (cathode ray tube) or a plain old television set. Monitors frequently have better quality pictures than household television and come in black and white, black and green, and color. Once the system has a monitor, the potential is there for the use of a light pen as an input device. Some students lack proficiency in their native language or typing skills, so programs can be written allowing user input by simply touching the pen to the appropriate spot on the screen.

The monitor is a flashy and efficient way to view computer output, but when its power is turned off no evidence of the results remains. Because of the frequent need for a written record, sometimes referred to as "hard copy," most micro systems incorporate a line printer. Line printers come in a variety of speeds and prices and are designed to satisfy different needs. A major distinction in the nature of the printing itself is whether it uses fully formed characters or creates characters by way of a dot matrix head. Fully formed characters are more likely to be of letter quality, but dot matrix printers can be taught to print graphics displays as well as text. Some companies promote a thermal printer, but common complaints include fading of the image, irreproductibility by photocopiers and the high cost of paper.

Two devices, a cassette tape player and a disk drive, serve the micro as both input and output. Programs can be recorded on tape or plastic record-like diskettes and then entered into the computer for processing. The procedure can

be reversed by a user who has a program to store: one can output that program either to tape or diskette for safekeeping. The program will survive there for an indefinite period of time independent of the presence or absence of current in the micro. The user should be aware that the disk method of saving and retrieving information is many times faster and is considered more reliable than tape. The discs also hold more programs than a cassette tape.

Because motivation is so important in education and because well done color graphics displays can enhance motivation, another hardware component which deserves attention is a graphics tablet. With such an aid interfaced with a micro, teachers and students can draw on the tablet and watch their creations appear on the screen. The user can choose colors, draw, fill, magnify and diminish drawings, and eventually save completed pictures for incorporation in a program.

A promising but as yet not widespread type of device is the voice input. Micros with this capability can be programmed to make oral statements to the student. Once perfected, the voice recognition process will increase the value of the micro in the classroom, especially in cases where students have visual limitations.

If the microcomputer user finds it necessary to coordinate programs with another user or with a mainframe, then a modem (modulator-demodulator) can be purchased. The standard telephone receiver is inserted into this device which helps receive and send signals to another computer.

Summary

The microcomputer is an amazing product of our sophisticated technology and fulfills an important function in education. This chapter has traced the development of microcomputers, introduced its components and hardware, and has provided reasons for the micro's popularity. Specifics about hardware and software are provided in later chapters.

Chapter 4

MICROCOMPUTERS IN EDUCATION

William H. Kraus
Wittenberg University
Springfield, Ohio

Computer Languages

Although a computer is a very sophisticated electronic device, it can perform tasks for us only if it has been programmed to do so. A computer program is a set of specific instructions that tell the computer, step-by-step, what it is to do. Such a program must be written in a language the computer understands. Commercial programs can be purchased or teachers can learn the language of the computer and write their own programs.

An example of a computer program is given in Figure 4.1. This program is written in BASIC, the programming language most commonly available on microcomputers. Each statement in a BASIC program is numbered and the computer interprets and

```
10 PRINT   "I will add two numbers for you."
20 PRINT   "What is the first number";
30 INPUT   A
40 PRINT   "What is the second number";
50 INPUT   B
60 LET C = A + B
70 PRINT   "The sum of"; A; "and"; B; "is"; C; "."
80 END
```

Figure 4.1. A Simple Computer Program

executes the statements in numerical order. In the program in Figure 4.1, line 10 tells the computer to print the statement "I will add two numbers for you." on the TV screen. Line 20 tells the computer to ask the person using the program (the "user") for a number. Line 30 tells the computer to take the value entered by the user and assign it to the variable "A". Lines 40 and 50 are similar to lines 20 and 30. Line 60 assigns the sum of the two numbers in A and B to the variable "C." Line 70 prints out the answer in a complete sentence. For example, if the two numbers supplied by the user were 37 and 49, line 70 would tell the computer to print "The sum of 37 and 49 is 86." Line 80 tells the computer that it has reached the end of the program.

Computer languages are typically very limited in vocabulary and very exacting in syntax. A small change in the above program as simple as changing the word "PRINT" to "WRITE" or accidentally using "PRITN" will cause the program to abort. Similarly, changing the order of the statements or changing, eliminating or adding punctuation will cause the program to abort or at least to function in a manner different from that originally intended. Although this may seem like a rather long program to do a very simple task, it is possible to change line 60 to do a very complex calculation which the computer will perform with its characteristic speed and accuracy.

There are many other computer languages; three in particular are worth noting. Pascal is the favorite language of

many experienced programmers because of the ease with which it can be used to construct large, complex programs out of smaller, simpler "modules." LOGO is a simple language developed at MIT especially for young children to use in writing programs that draw colorful geometric figures on the TV screen. PILOT is a language designed for use by teachers in creating computer programs for use in instruction.

Somewhere in the above discussion about computer languages you may have thought to yourself, "But I thought computers had something to do with binary arithmetic." If so, you were right. All computer programs must be translated into a set of instructions to the computer expressed as binary numbers; this is usually referred to as machine language. Figure 4.2 gives machine language code (for microcomputers that use a "6502 chip") that is equivalent to the

10100101	Load the accumulator with the number found in the following memory location.
11001011	Address of a memory location used for storage.
00011000	Clear the carry (in case there is a carry left over from a previous operation).
01100101	Add the number found in the following memory location to the number in the accumulator and store the result in the accumulator.
11001100	Address of a memory location used for storage.
1000101	Store the number from the accumulator in the following memory location.
11001101	Address of a memory location used for storage.

Figure 4.2. Machine Language Code With Explanation

single BASIC statement "LET C = A + B" (line 60 of the program in Figure 4.1). If this little bit of machine language seems obscure, consider that adding numbers larger than 255 or multiplying numbers would be much more complex and that to write the entire program from Figure 4.1 in machine language would be a very complex task.

Because writing programs in machine language is difficult and time consuming, most programs are written in "higher level" languages such as BASIC, FORTRAN, COBOL and Pascal. These languages permit use of letters, symbols and mnemonics to produce programs which are easier to read and to understand the logic.

BASIC is the most widely available language for microcomputers and programming in BASIC is taught at both the elementary and secondary levels. Pascal is another popular language for microcomputers. Although Pascal requires a large amount of computer memory and usually a disk drive as well, it will probably become a commonly taught language at the secondary level because of its growing use in many branches of computer science.

Computer Assisted Instruction

One of the most important uses of microcomputers in education is for computer assisted instruction (CAI). There are several categories of CAI; in each the student learns directly from the computer.

The simplest and most common instructional use of the computer is to provide the student with <u>drill</u> on content the

student has previously learned but not yet mastered. For example, in Fish Chase (Kraus, 1980) a big fish chases a little fish across the TV screen while the student attempts to answer 20 addition or multiplication problems before the big fish catches and eats the little fish. Many features of microcomputers make it possible to provide effective drill. The features include these capabilities: to randomly generate problems; to adjust the sequence and difficulty of content based on student performance; to provide or withhold feedback as appropriate; to include a time factor; to combine symbolic, auditory and pictorial stimuli (including animation); and to keep records of student performance for the teacher.

A second category of CAI is the tutorial in which the student learns new content from the computer. The features of the microcomputer that can be used to enhance drill can also be used to enhance tutorials. Although good tutorials have existed on large computers for many years, it is much more difficult to find good tutorial programs than good drill programs. The PLATO system is an extensive computer-based education system developed at the University of Illinois and marketed through Control Data Corporation, but it requires a larger computer. Most current tutorial programs available for microcomputers are mundane text-oriented programmed learning. (Such programs are sometimes referred to as "page-turners" since the computer does little more than turn pages in a programmed-learning textbook.) At present, great care

must be taken in selecting tutorial programs for use in the classroom.

A third and particularly enriching category of CAI is simulations. A simulation is a simplified model of a real-world situation. For example, in Lemonade the student runs a simulated lemonade stand. Given a weather forecast and the costs of ingredients for lemonade and of material for advertising signs, the student must decide how much lemonade to make, how many signs to put up and what to charge for a glass of lemonade. Based on the student's decisions and on the weather, the computer determines how much profit, if any, the student makes. Random events such as street repairs (since the street is closed only three glasses of lemonade are sold) and thunderstorms (the stand is demolished) are included in the program. A similar program, Market, allows several companies to compete with each other for profits. Other simulations involve topics such as city planning, government and ecology. A good simulation is a fascinating instructional tool that can do an excellent job of relating learning to real life.

A fourth category of CAI is demonstrations or programs used by the teacher to illustrate certain points in a lecture, much as a transparency or slide or movie might be used but allowing an extra degree of interaction with the media. For example, in discussing conic sections a teacher might use a computer program that accepts parameters such as eccentricity and distance between focus and directrix and then gener-

ates the corresponding hyperbola, parabola or ellipse on the TV screen. By overlaying various conic sections on the screen the teacher can demonstrate a variety of relationships, and if a student asks "But what if...?," the teacher can supply new values to the computer to answer the student's question.

Although not a separate category of CAI, games are a very important instructional tool. There are many effective drill and simulation games. The effectiveness of instructional games has been demonstrated many times (e.g., Bright, Harvey, and Wheeler, 1979; Kraus, 1980). Combining the motivational aspects of games with the instructional capabilities of microcomputers creates an instructional tool with a wide variety of applications (See Malone, 1980, for an interesting study of computer games.) One word of caution in the case of games: the word "game" has negative connotations for many people who equate it with "frivolous." If you are careful to select games that have clear instructional objectives, you will find it much easier to justify their use in your classroom. Although some teachers would argue to the contrary, noninstructional computer games like Space Invaders or Pac-Man have little or no place in the classroom. Even when you wish to use a computer game as a reward for certain behavior or achievement in your class, you will find that instructional games like Lemonade, Fish Chase and Hangman are greatly enjoyed by students.

Computer-assisted instruction is easily introduced into a classroom. A single microcomputer will allow drills, tuto-

rials, simulations, demonstrations and games to be used to increase and enrich learning in many areas of the curriculum and to give students experience in using computers. Unlike instruction in programming, the use of CAI requires almost no special training on the part of the teacher.

Other Instructional Computing

Computer-assisted instruction and instruction in programming are the two dominant classroom uses of microcomputers. However, there are a number of other instructional applications of microcomputers.

Computer-managed instruction (CMI) has been used in many school districts across the country for over a decade. In CMI the student does not receive instruction from the computer, but instead the computer is used to manage the student's instruction. The computer keeps records of individual student's progress and tells the student or the teacher what assignments the student is to work on next. Individualized instruction is facilitated by CMI, with the computer relieving the teacher of much of the tedious paperwork involved. Some large systems, most notably PLATO, combine computer-managed and computer-assisted instruction to create an instructional environment that requires little teacher intervention.

Because large time-sharing computers have greater capacity for storing programs and data than microcomputers and because they can communicate simultaneously with large numbers of users, they are generally better suited to CMI than

are microcomputers. Nonetheless, CMI is available for most microcomputers, most often in the area of mathematics, and more will become available in the next few years. It is important for teachers and administrators to note that adopting a CMI system in a classroom or district often involves a high degree of commitment to the system. Adopting an extensive CMI system is not only similar to adopting a curriculum but it also can involve transferring control of the curriculum from the teacher to the computer. Because of this some school districts have developed their own CMI systems tailored to meet their specific needs. Unless a school district is fully committed to adopting or creating an extensive CMI system, it is probably best to adopt a simpler CMI system that will assist the teacher with planning, evaluation, and record-keeping while the teacher maintains the flexibility to develop a sound educational program ("teacher-managed" instruction).

There are a number of potential classroom applications of microcomputers that are awaiting development of good software. One of the simplest is to have students use the computer as a tool in the same way that computers are used in business and industry. For example, the major business uses of computers are for data-base management (storing and retrieving information) and text processing (creating and modifying letters, reports and manuscripts). Simplified versions of these programs that would allow students to keep records and produce reports about the weather, hobbies, sports, TV

programs and other things of direct interest to them would be extremely useful in creating computer literacy in students. Other ideas just developing are networks (in which students and teachers can use a microcomputer to communicate with other students and teachers and to tap a variety of other resources via telephone lines) and the combining of microcomputers with video-disc players to create instruction that capitalizes on the "intelligence" of computers and the image-producing capabilities of television.

Getting Microcomputers into the Classroom

Computers need to be introduced into classrooms in every school immediately for at least two reasons. First, computer use is spreading throughout society making it more and more important to our society as a whole and to the individuals within our society that every citizen be literate and comfortable with computers. Computer literacy may someday soon be as important as reading, writing and arithmetic. Second, computers can be a useful tool in the educational process. Just as other sectors of our society use this tool to be more effective, so should our schools. The cost, portability, and sound and graphics capabilities of microcomputers make them more cost-effective for most educational applications than the time-sharing computer systems of a decade ago.

Because microcomputers have been used in only a limited number of schools for only a few years, evidence on the best ways to organize computer use in the classroom is only beginning to emerge. At the elementary level the computer can be

used effectively as a learning center in the classroom. All of the features of good learning centers can be programmed into a computer: motivation, feedback, sequencing, individualization, timed practice, tutorial aid and recordkeeping. At the secondary level a combination of a few computers available for use in individual classrooms and a computer lab with 10-25 computers may be best. Although the ultimate goal is for every student in the classroom to be seated at a computer at once, currently one computer per elementary classroom and a ratio of one computer for every 25 students at the secondary level seems to be a reasonable short-term goal.

Many teachers who have used computers with their classes have noted that students work well together at the computer even with programs that were specifically written for individuals to use. Students also work well in tutoring other students in the use of the computer or in programming.

The interest and the variety of educational experiences that can be generated by a microcomputer must be seen to be fully appreciated. Any school without a computer should make every effort to get one whether through normal channels or through the PTA or fund raising events or even by borrowing one from a dealer, a college, a computer club or a parent. The first computer in a school can generate the interest (and the pressure on the school board) to obtain more computers, software and training for the teachers.

References

Ahl, D. H. Computer Simulation Games. Teacher, 1980, 97 (5), 60-61.

Allen, M. Computer-managed Instruction. Journal of Research and Development in Education, 1980, 14 (1), 33-40.

Bork A , and Franklin, S. D. The Role of Personal Computer Systems in Education. AEDS Journal, 1979, 13 (1), 17-30.

Bright, G. W., Harvey, J. G., and Wheeler, M. M. Using Games to Retrain Skills With Basic Multiplication Facts. Journal for Research in Mathematics Education, 1979, 10, 103-110.

Burton, R. R., and Brown, J. S. An Investigation of Computer Coaching for Informal Learning Activities. International Journal of Man-Machine Studies, 1979, 11, 5-24.

Carpenter, T.P., Corbitt, M. K., Kepner, H. S., Lindquist, M. M., and Reys, R E. The Current Status of Computer Literacy: NAEP Results for Secondary Students. Mathematics Teacher, 1980, 73, 669-673.

Dence, M. Toward Defining the Role of CAI: A Review. Educational Technology, 1980, 20, (11), 50-54.

Dickerson, L., and Pritchard, W. H. Microcomputers and Education: Planning for the Coming Revolution in the Classroom. Education Technology, 1981, 21 (1), 7-12.

DiGiammarino, F. P. Computers in Public Education: The Second Time Around. Catalyst for Change, 1980, 10 (1), 8-11.

Doorly, A Microcomputers for Gifted Microtots. G/CIT, 1980, 14, 62-64.

Eisele, J. E. A Case for Computers in Instruction. Journal of Research and Development in Education, 1980, 14 (1), 1-8.

Johnson, D. C. Computer Literacy - What is it? Mathematics Teacher, 1980, 73 (2), 91-96.

Kraus, W. H. Using a Computer Game to Reinforce Skills in Addition Basic Facts in Second Grade. Journal for Research in Mathematics Education, 1981, 12, 152-155.

Lubar, D. Educational Software, Part II. Creative Computing, 1980, 6, (10), 56-60.

Malone, T. W. What Makes Things Fun to Learn? A Study of Intrinsically Motivating Computer Games (Cognitive and Instructional Science Series, CIS-7, SSL-80-11). Palo Alto, CA: Xerox, 1980.

Miller, I. The Micros are Coming. Media and Methods, 1980, 16, (8), 32-34, 72, 74.

Milnar, S. D. Teaching Teachers About Computers: A Necessity for Education. Phi Delta Kappan, 1980, 61, 544-546.

Molnar, A. R. The Coming of Computer Literacy: Are We Prepared for it? Educational Technology, 1981 (1) 21, 26-28.

Rawitsch, D. G. Implanting the Computer in the Classroom: Minnesota's Successful Statewide Program. Phi Delta Kappan, 1981, 62, 453-454.

Schuyler, J. A. Programming Languages for Microprocessor Courseware. Educational Technology, 1979, 19 (10), 29-35.

Wiebe, J. H. BASIC Programming for Gifted Elementary Studdents. Arithmetic Teacher, 1981, 28 (7), 42-44.

Wieck, C. Computer Resources: Will Educators Accept, Reject, or Neglect in the Future? Education Unlimited, 1980, 2 (3), 30-34.

Zukas, W. X., Berka, L. H., and Martin, J. T. Teaching Fourth and Fifth Graders About Computers. Arithmetic Teacher, 1980, 28 (2), 24-27.

Chapter 5

SELECTING MICROCOMPUTER SYSTEMS: A PROCESS MODEL

Pierre P. Barrette
Southern Illinois University
Carbondale, Illinois

In this chapter we examine the use of a process model to select microcomputer systems (as defined in Appendix A). We also examine selected features common among different microcomputers and attempt to relate these features to applications for science and mathematics teachers and students.

Selecting a Microcomputer

It must be stated categorically at the outset that no empirical research evidence exists to suggest that microcomputer brand A is better than microcomputer brand B for all purposes. Certain individuals and groups will favor one machine over another for different reasons. There are many good microcomputers currently available and the new models being designed are even better.

The actual process of selecting a microcomputer is not as difficult as one would imagine. Obviously, one must learn new terms and concepts in order to be able to make intelligent comparative assessments. Surprisingly, the easiest step is to ultimately compare and make a choice of a particular machine. The most difficult part comes earlier. This part is associated with answering a number of questions about how a particular machine is to be used in the teaching/learning process. The more time spent on addressing the question of

how the machine is planned to be used, the easier the ultimate purchase decision will be. It should be remembered that there is no "final" choice in terms of microcomputers or microcomputer systems. The machines currently available will become outdated and will have to be replaced.

There are many different approaches to selecting a microcomputer system. One can expect to receive all types of information from various people including colleagues, hardware vendors and software vendors. Everyone has a set of personal reasons why one should buy a particular brand or model. Information one receives can be conflicting or completely erroneous. Be prepared to receive rumors about the weaknesses of one brand as opposed to the strengths of another. The virtue of objectivity has not become enhanced with the advent of microcomputers. In spite of this, one can proceed to make informed decisions about microcomputers by using a process model involving as few as seven major steps. (Figure 5.1)

Using the Process Model

The first step in the process of selecting a microcomputer system is to create a written list of potential applications. This list should contain a series of statements about how the system is to be used. A list of mathematics and science applications could include items from this list.

1. All planned uses that may involve students as they employ drill and practice, tutorial, simulation, instruction-

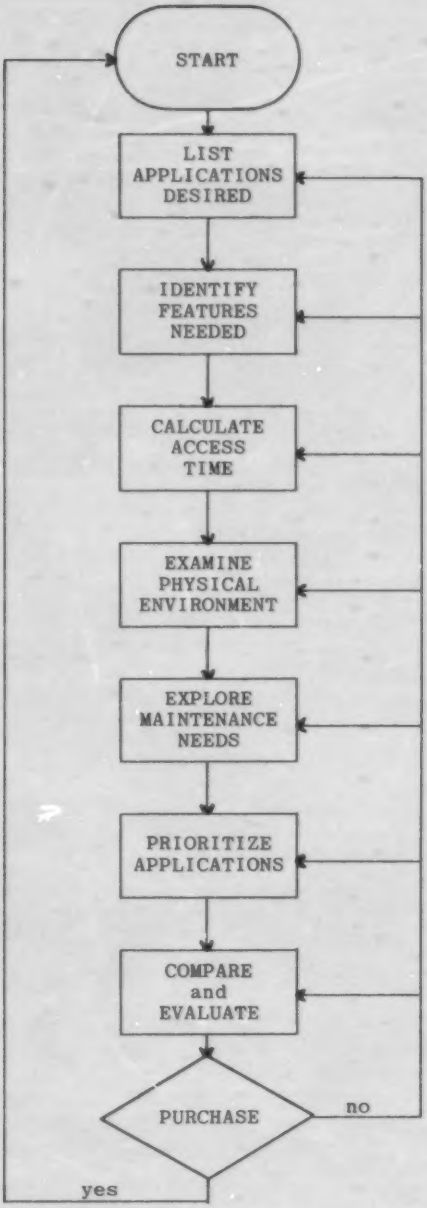


Figure 5.1. Microcomputer Selection Process Model

al games or other types of previously prepared program materials on a one-to-one basis.

2. All planned uses that may involve students on a one-to-one basis as they are involved in using microcomputers to acquire language or computer process skills involving any of the high or low-level computer languages.

3. All planned uses that may involve students utilizing microcomputers for monitoring, recording and analyzing real time events. These real time events are normally associated with conducting experiments in which analog to digital conversions are made by the microcomputer.

4. All planned uses by the teacher in which a microcomputer may be used for classroom demonstrations of any kind as opposed to single one-to-one student use.

5. All planned uses by the teacher in which a microcomputer system may be used to facilitate the analysis and assessment of student progress.

6. Other uses for locally producing materials, instructional management or other applications not previously described.

Your initial list is not likely to reflect all the applications that are possible as you will discover new ones as time goes on. An excellent list of specific microcomputer based instructional applications for Algebra I, Algebra 2, and Geometry can be found in the Title IV-C Iowa Microcomputer Curriculum Project. (See Appendix B)

Regardless of the subject or grade level, the important

first step is to write down each potential application separately. This first step is facilitated by the use of small cards. Using small cards with headings printed on them saves time as well as being very helpful in finally prioritizing the ultimate choices. Use a single card for each application.

```
MICROCOMPUTER APPLICATION                    PRIORITY #

Computer control process skill curriculum develop-
ment introduction.
Six students.  Grade 5
For 36 weeks at 2 hour/week/student/

-------------------------------------------------------

FEATURES NEEDED                              ACCESS TIME

- 48 K RAM                                   One to One
- B & W MONITOR                              12 hours/week
- 2 FLOPPY DISK DRIVES
- LOGO LANGUAGE CARD OR ROM/pack

ENVIRONMENT.  Math Lab
```

Figure 5.2. Sample Application Card

After writing down an individual specific application, you should list the microcomputer system features that may be needed in order for this application to be realized. An examination of selected microcomputer features and a table of features found in selected microcomputers follows in this chapter.

After writing down the application and the features needed for that application, examine the question of access

time very carefully Access time is the actual amount of either one-to-one student contact time with a microcomputer or the contact time needed by a teacher for direct instructional presentations or other uses. Unfortunately, access time has seldom been considered in the selection process although it is likely to be one of the most important factors in determining the success or failure of implementing a microcomputer based project. One may be able to double or even triple access time per dollar by knowing clearly the features needed for planned applications and by purchasing the brand of microcomputer or microcomputer system to accommodate the needs.

The fourth step in the selection process is to examine the various rooms and physical environments where the applications are likely to take place. Without question, a great many applications will take place in the classroom. Special rooms can be set up for multiple microcomputer systems but this may pose additional staffing questions as well as the possibility of student control and scheduling problems. Aside from the classroom, one of the best locations for multiple microcomputer systems is in the media center. Reserving and circulation of materials can be arranged with the media specialist. If there is a limited number of microcomputer systems and access time is anticipated to be limited, you should consider the need for student sign-up schedules.

Other physical environment considerations include the

need to have sufficient grounded electrical power. Damage to microcomputers can occur if the power surges or drops because of other heavy power consuming equipment. Also, power surges or drops may interrupt or crash programs, an event which is frustrating and irritating to the user. Be sure that the microcomputers are not on the same electrical circuit that is being used for televised instruction. Microcomputer generated interference can affect the quality of signals being received on television sets near them. Static discharges from users traveling across carpeted surface areas, especially in warm or low humidity conditions, can easily cause programs that are in use or being written to crash.

The fifth step in the selection process is associated with the issue of how maintenance service for microcomputer systems is to be provided. Ninety days is approximately the length of most warranties on microcomputers and system peripherals. You can anticipate a minimum average of five percent system failure time for each piece of equipment per year. As you list and examine the features needed, ask yourself, "Who is going to perform the periodic maintenance on the equipment?" and "Who is responsible for repairing the equipment when it breaks down?" Larger school districts have centralized media equipment repair programs. If your school district does not provide for central media equipment maintenance you will need to obtain maintenance from outside sources. One should estimate the maintenance cost to be approximately 10 percent of the capital cost of the equipment

per year. This cost may or may not include the cost to upgrade the microcomputer systems as manufacturers come out with new enhancements. One should not expect an overly friendly reception when looking for outside maintenance from a vendor near you who did not receive the bid for equipment.

Service contracts are available for most microcomputer system equipment. Generally a service contract costs about 10 percent of the retail cost of the equipment per year. Service contracts vary widely and should be carefully examined to find out what they do or do not contain. Failure to anticipate service maintenance needs associated with microcomputers can seriously affect the implementation of a program no matter how finely stated each of the applications are.

Step six in the selection process is to sort the applications you plan and to prioritize them. Sort on the basis of what you consider to be the most important applications that can realistically be accomplished considering the access time available. Do not overstate what you can reasonably accomplish with the anticipated equipment and the time needed to integrate the planned applications into the curriculum. It takes much time to implement a project even with only one or two microcomputers. Save or use the additional lower priority applications to build a case for additional systems.

The final step is to compare the features of various brands of microcomputers with the features desired to accomplish the applications. When doing the actual comparisons,

be sure to ask vendors if the particular model you may be considering is the latest model. Compare several brands. If possible, ask individuals who own a particular piece of equipment what their experiences have been with the equipment and with any service problems. Remember to focus on the features you desire for yourself and your students. Once the decision is made, consider the possibility of group or network purchasing and how this relates to providing service.

Important Features to Consider

The following is a list (Table 5.1) and brief description of microcomputer features that one may want to consider. It should be understood that any statements made about microcomputer features or microcomputer systems in this chapter are dependent upon what was available at the time of this writing.

Microprocessor

The heart of any microcomputer is its microprocessor. For most school science and mathematics applications that rely on the use of a microcomputer to deliver instruction by running prepared or purchased programs, the question of which microprocessor one should have is usually not important. Unless one has specific applications that require a particular 8 or 16-bit (Appendix A) microprocessor, one need not be overly concerned whether the resident microprocessor is an 8 or 16-bit slice machine.

Color

Whether color is an important feature to have is often a

Table 5.1

MICROCOMPUTER FEATURES

MODEL	Source	Suggested list	Microprocessor	Bit size	Color capable	User available RAM	Number of keys	Available ROM	ROM/RAM Cartridge	Cassette Included	Floppy Disk
MODEL "R"	a	$ 24[illegible]	8080A	8	Y	16K	53	2K	N	Y	NA
VIC 20	b	300	6502A	8	Y	3.5K	66	16K	Y	*	*
TRS-80 Color	c	400	6809A	8	Y	4K	53	8K	Y	*	*
Challenger C1-P	d	479	6502	8	N	4K	53	10K	N	Y	*
TI 99/4A	e	595	9900	16	Y	16K	53	26K	Y	*	*
Atari 400	f	630	6502	8	Y	8K	57	10K	Y	*	*
TRS 80 Model III	c	699	Z80	8	N	4K	65	4K	N	*	*
Commodore 2001	b	795	6502	8	N	8K	64	14K	N	*	*
Challenger C4-P	d	879	6502C	8	Y	8K	53	10K	N	*	*
Challenger C8-P	d	895	6502A	8	Y	8K	63	10K	N	*	*
Atari 800	f	1080	6502	8	Y	16K	57	10K	Y	*	*
Excidy Sorcerer	g	1295	Z80	8	N	16K	63	4K	Y	*	*
Heath 88	h	1295	Z80	8	N	16K	72	8K	N	*	*
Apple II+	i	1330	6502	8	Y	16K	52	8K	N	*	*
Bell & Howell	j	1420	6502	8	Y	16K	52	8K	N	*	*
IBM Personal	k	1595	8088	16	Y	16K	83	40K	N	*	*
Commodore 8032	b	1795	6502	8	N	16K	73	14K	N	*	*
Osborne 1	l	1795	8080A	8	N	60K	68	4K	N	*	2
Compucolor II-3	m	1895	8080A	8	Y	8K	--	16K	N	*	*
Heath H-8	h	1970	8080A	8	N	16K	73	1K	N	*	*

*Optional. Note: Prices are manufacturers suggested retail prices in 1981. Actual prices vary extensively depending upon quantity and special promotions.

judgment based upon an individual's preference. It is akin to asking whether a color or a black and white TV is needed. At this time, little or no empirical research evidence exists to suggest that color produces significant learning changes for students involved with microcomputer based programs. Of course, if color is the object of a program, then the argument does not hold. If previous comparative research involving color in film, television and slides can be an indicator, it is likely that color associated with microcomputers will also yield findings of no significant difference. It does not make much sense, however, to insist on color as a microcomputer feature and to use color based programs, but then equip the microcomputer with a black and white monitor to "save" money. One can purchase a very fine microcomputer and black and white monitor for less than the cost of a color monitor itself. One can also purchase a very fine microcomputer capable of generating color and not use the color feature at all since other features are more important.

User Available RAM

This feature relates to the actual amount of user available read/write memory that a microcomputer has. This is an important feature to consider. The emphasis is upon the term "user available" as opposed to the actual stated amount of read/write random access memory (RAM) that is often present. The electronic size of a random access read/write memory is currently measured in K byte units. (See Appendix A) A microcomputer with an 8-bit microprocessor can be equipped

with various sizes of addressable read/write RAM memory ranging up to 64K. One must purchase additional read/write RAM chips or plug-in read/write RAM packs to increase the size of the memory for many desired applications.

The actual amount of "user available" read/write memory in K units is often considerably less than the RAM that the machine actually possesses. This is because part of the read/write RAM memory is used to store special computer programs that make the microcomputer operate. When examining the "user available" RAM feature, ask the question, "How much user available read/write RAM memory will I need?" If you don't know, then it may be wise to purchase as much read/write RAM memory as possible even with the first microcomputer. Ask vendors how much user available read/write RAM memory is actually available even though you may have purchased as much memory as is possible with a particular brand. As you compare various brands, pay particular attention to the amount of read/write RAM memory that must be occupied by different high-level computer languages. Consider also the later comments associated with ROM/RAM cartridges when examining user available RAM.

Keyset

The keyset of a microcomputer is one of its most important features. The term "keyboard" is often used instead of keyset. Very few microcomputers have precisely the same set of keys even though they may have a similar number of keys. Keys needed on a microcomputer depend very much on the appli-

cations one has in mind. When examining the keyset of a microcomputer, consider the following questions and comments:

1. Does the native keyset include a full ASCII (Appendix A) character set without a hardware modification or software enhancement? You may need a full ASCII character set if you plan to involve yourself with local production of materials involving upper and lower case characters, square brackets and other features. Upper and lower case and full ASCII adapter chips can be purchased at additional cost on many microcomputers.

2. Does the native keyset include a numeric keypad? Applications with extensive amount of numeric input may need this feature. With some systems it is included. With others it is optional.

3. Does the native keyset provide for user definable keys? User definable keys are special individual keys that permit one to define how they will operate when pressed. Normally, a user definable key can be programmed to generate a special character or graphic, or can represent a full predefined computer procedure. Applications for user definable keys include the ability to define complete equations with a single keystroke, to create output with numeric or alphabetic subscripts and superscripts, and to generate graphic sigmas. In addition, user definable keys can be used for hexadecimal entry.

4. Does the native keyset include graphic keys? This feature may be particularly important in order to prepare

monitor graphic images from the keyboard. Some keysets place actual images of partial graphic symbols on the vertical portion of their keys. Look closely at your applications to determine if the need to prepare graphic images is an important feature and whether or not a particular brand of microcomputer has graphic keys as a standard offering.

5. Examine how many noncontrol type keys on the keyset are individually repeatable with a single keystroke.

6. Examine closely the native keyset to determine if cursor control keys are clearly identified.

7. Check to see if program interrupt keys or break keys can be disabled manually as well as under software control.

8. Finally examine the actual tacticity of the keys including whether or not fingers may accidentally slip from one key to another. Few keysets on microcomputers have been designed with human factor ergonometrics in mind. This is unfortunate as the keyset of a microcomputer is likely to be the major microcomputer-human interface in the near future.

Available ROM

ROM is the acronym for Read Only Memory, the special random access memory of a microcomputer. The user can only retrieve "read" information stored in this type of memory and cannot store information into it. (More sophisticated memory units are identified in Appendix A.) Often, manufacturers place their high-level language interpreter or other computer codes into read only memory chips. The size of a microcom-

puter's ROM will vary from one manufacturer to another and is measured in K units similar to read/write random access memory RAM. If possible, have the vendor explain what the read only memory of a microcomputer does.

ROM/RAM Cartridges

Often, ROM chips are packaged in separate external containers called cartridges or command modules. The capability to externally plug into a microcomputer additional ROM or RAM is available on several models. Some plug in cartridges contain both ROM and RAM chips. ROM cartridges can contain any number of computer programs such as a different language, an instructional program, a management program or a communications program. Any program stored on a floppy disk can be prepared for permanent storage into a ROM cartridge. An advantage of a read only memory cartridge is that there are no moving parts and one may store a rather large and sophisticated program in it and not need a correspondingly large read/write RAM in the microcomputer.

Cassette Decks

Cassette storage decks for microcomputers employ an actual or slightly modified audiotape cassette recorder/player. There are special digital cassette decks as well as digital cartridges, but these are not discussed here. Cassette decks are a very popular feature with microcomputer users because of their low initial cost compared to floppy disk drives. Some brands of microcomputers include a cassette deck built into the chassis while others provide only a

receptical in order to plug in an external deck. Normally, programs and data cannot be loaded or saved as quickly on cassette decks as on floppy disks. However, initial costs and other considerations may compel one to elect a cassette deck for a storage device.

One may also want to consider another type of storage device called a stringy floppy. This storage device employs a long strip of magnetic tape in an endless loop cartridge that is similar in principle to an audio cartridge. A stringy floppy costs more than a cassette deck but considerably less than a floppy disk drive. It has high K storage, very rapid load and save time and has enjoyed a reputation of being very reliable.

Floppy Disk Drives

Floppy disk drives that use 5¼ inch diskettes are generally available only as optional features for most microcomputers. One of the microcomputer systems listed on the features table, however, included two floppy disk drives as part of their basic system. The operation of floppy disk drives vary from one manufacturer to another and as a result are not compatible with each other.

Floppy disks can store large amounts of information as measured in K units. One must, however, be careful to ask the question, "How much user available space is there on the first, second and subsequent diskettes?" Many floppy disks contain a computer program on the first diskette called an operating system. This operating system may use up to one-

half or more of the available storage space on the first diskette leaving little user available storage space.

Floppy disk drives are classified as being either single density, double density, double sided or double sided double density. Single density disk drives permit thirty-five or more concentric storage tracks on one side of a diskette. A double density disk drive literally doubles the storage density on a track or increases the number of tracks on one side. Double sided means that both surfaces of the diskette are used.

Floppy disk drives are electromechanical devices and as such are more susceptible to failure than cartridge storage devices. Heads of a disk drive need to be cleaned as part of a periodic maintenance program. Training in care of handling of diskettes by students and other users will greatly assist in reducing failures due to damaged or contaminated diskettes.

Hard Disk Drives

Hard disk drives that use a 5¼ inch or a larger sealed disk are becoming more popular as an optional feature. The advantage is one of being able to store larger amounts of information, often measured in megabytes (MB). A typical single 5¼ inch hard disk drive system will cost four or more times that of a single floppy disk drive system and also will be able to store ten or more times what a single floppy disk can store. Some single platter hard disk drives available for microcomputers have storage capacities in excess of 30 MB.

When one is considering whether to purchase a hard disk system, the actual physical environment where the hard disk system may be located should be carefully considered as contaminants, sharp jolts, or power failure can cause serious damage. The vendor can explain further. Finally, the real cost advantages of having a hard disk drive system together with the need to purchase other machines to back up the information stored on the hard disk should be determined in advance.

High Level Languages

Microcomputer manufacturers normally supply at least a single dialect of BASIC as the high level language with each system they sell. Dialects among high level languages offered by different manufacturers are not usually compatible with each other. This is particularly true in the area of graphic commands. Other high level languages can be purchased from either the original manufacturer or some other vendor. These high level languages include Pascal, PILOT, LOGO, LISP, FORTRAN, FORTH, COBOL, APL, Smalltalk and others.

The UCSD Pascal language is becoming increasingly popular in secondary math and science because of its structured approach to learning how to program. The language is also particularly noteworthy as a high level language because its design permits it to be the most machine independent high level language developed to date. Machine independent implies that programs created on one brand of microcomputer will also operate on various other brands of microcomputers.

This is particularly important when one takes into account the exchanging of application programs among microcomputer systems users. One can expect computer languages of all types to evolve with enhanced features involving graphics, speech and image recognition in the next few years.

Operating Systems

Operating systems are special computer programs that manage the storage and retrieval of other computer programs and data from the read/write RAM or read only memory. A microcomputer that offers disk storage devices as optional features will also provide at least one operating system. Any applications that require extensive storing and retrieving of information from disks become very dependent on the capabilities and limitations of various operating systems. If you are not or choose not to become conversant with the features of various operating systems, it would be most appropriate to spend additional time speaking with different people to understand this most important feature.

Low Level Languages

For some instructional applications, the need for a low level language feature such as an assembler exists. This is particularly true when teaching assembly level programming. For the most part, assemblers are not included in the normal vendor price even though machine level access may be available. The architecture desired and instruction set available become important microcomputer selection considerations especially as they relate to how extensive the available

written documentation for the system is. Vendors and manufacturers should be consulted directly for information. Critiques through periodical articles help in this effort.

Printers

Few microcomputer manufacturers offer printers as part of their basic system. Printers are typically produced by other companies. Some microcomputer manufacturers are beginning to offer the interfaces (the physical and electrical connections) needed to make a connection between a microcomputer and a printer as part of the basic cost.

Printers come in either impact or nonimpact varieties, and there are many models of printers. As a minimum, you may want a printer that will print all of the ASCII characters and the graphic characters your microcomputer generates. One should include the ability of the printer provide a black and white printout of whatever is displayed on the monitor. Several printers with improved features have become available at reasonable costs.

Monitors

Examining the microcomputer features table, one can quickly determine if a monitor is included in the purchase price. Monitors vary extensively in quality especially in the low cost, black and white variety. Green phosphorus screens on monitors are desirable for those who make extensive use of a monitor for displaying text information or with certain high-level languages.

One should carefully compare at least two different mon-

itors using the same microcomputer and the same program before selecting the monitor. If one plans to employ an eighty column microcomputer display, it becomes increasingly important that the monitor be able to operate in the frequency range of 12 MHz for high quality resolution of the character information that is displayed.

Some microcomputer manufacturers include or provide optional adapters to permit the use of a regular television receiver. These RF adapters generally cannot provide as clear and distinct an image as those with a direct electrical video connection as is found in monitors.

Text Characters/Lines

Microcomputers have different capabilities to display text characters on monitors. One normally measures the number of characters that can fit onto a single horizontal line and then the number of possible lines that can be displayed on the screen. The table includes only the first level of text characters per line. However, be sure to examine the readability of all available levels of displayed characters especially if both upper case and lower case with descenders are included. Different high level languages and different types of programs to help one design or author lessons permit various combinations of shapes of characters, number of characters per line and number of lines.

Graphics Resolution

The ability of a microcomputer to faithfully reproduce an image is at this time somewhat limited. This limitation

is due in part to whether the microcomputer uses an 8 or 16-bit microprocessor, the amount of read/write or special video RAM memory space available for graphics, the performance features of a device called the video ROM and the frequency response of the monitor. One can expect much higher quality graphic images to be generated with a 16-bit processor. Most microcomputers have the ability to generate one or more levels of graphic quality excluding the level used for text characters. Some microcomputer manufacturers have recently introduced new high level language commands that easily permit graphic animation.

Examine various microcomputer models you may be considering together with the high-level language you anticipate using to see if you can mix both graphics and text information in the same area on the screen. Examine again the question of how much user available read/write RAM memory you will have actually available for the nongraphic parts of a program. If you anticipate graphics to be an important feature for locally produced programs, this will help you select the appropriate microcomputer.

Power Up Diagnostics

The final feature in this selected list is the availability of a resident power up set of diagnostics with the microcomputer. Power up diagnostics include special read only memory computer programs that the manufacturer includes as part of the basic microcomputer. The purpose is to provide the user with an electronic diagnosis of the microcom-

puter system to ascertain if the system is ready to be used. Simpler versions of these programs require only a few seconds to run and provide important system status information if there is a problem encountered.

There are other features of microcomputers that could be discussed and may be needed in the applications you define. These may include the presence of sound generated from a microcomputer, the availability of special slots or ports to plug in additional cards or parts, and the ability to interconnect microcomputers to form a local network. If these and other features are desired, one should check their availability and cost and compare models from several different vendors and manufacturers.

Conclusion

It is important to remember that microelectronic technologies that form the basis of microcomputers and microcomputer systems are in a rapid state of change. Features that are optional today are very likely to become standard with little or no additional cost. There are many good microcomputers available today and the ones coming are even better. By following the process model carefully, one should be able to systematically acquire and implement microcomputer systems in the school science and mathematics curriculum even though various microcomputer features change over time.

Chapter 6

MICROCOMPUTER SOFTWARE

Henry S. Kepner, Jr.
University of Wisconsin
Milwaukee, Wisconsin

The amount of use of microcomputers in classrooms over the next decade will be closely related to the software developments in educational computing. While modifications in equipment will continue, microcomputers are reliable, inexpensive and flexible enough for schools to put at least one in every classroom by 1987. Construction of software is the key to bringing about this inclusion of microcomputers.

Software

Software is a sequence of written instructions (a program) stored in a form that can be understood and acted upon by a particular computer. Unfortunately, the coding of one particular program usually permits execution of that software on only one model of microcomputer. Modifications would have to be made in the software before that program could run on other computers. Thus software for the TRS-80 will not run on the Atari computer; software for the TI-99 will not run on the Commodore PET; and software written for an Apple II with 48K may not run on an Apple II with 32K.

With this incompatibility of even those computers which claim to use the same language, the educator must know the details about a piece of software. These details should be included in the documentation for that software. The docu-

mentation should present the educational purposes, intended uses, resource materials and equipment requirements for the software. Examine each of these in relation to your situation. For now, do not purchase a software package unless it was written for your computer.

For microcomputers, software is typically available on cassette tapes or diskettes. The larger number of programs that can be stored on a diskette make it a better investment. The initial hardware expense of a $30 cassette player or a $500 disk drive favors cassettes. However, the diskette and disk drive are recommended for most instructional settings.

From the educational view, software programs contain selected course content, instructional design and procedures for interacting with the learner. From this perspective, the examination of computer software is more important than comparable evaluations of textbooks, drill materials and other aids. To be of instructional value to the teacher, the software must be able to interact with individuals or small groups with limited teacher intervention. Since computer lessons are conducted under basic control of the software, rather than teacher or learner control, the need for thorough evaluation is critical.

In seeking computer software, an educator must know what the computer is expected to do. When computer software is used without clear goals and evaluations, everyone is dissatisfied. Frequently, the dissatisfaction results because the software generates learning that differs from the teacher's

expectations. Before addressing the questions of expectations and evaluation in detail, let us find out more about software available today.

Finding Software

Where do you find educational software? The answer is, EVERYWHERE! Educational software is being developed in basements, bedrooms, garages, school rooms and business offices. Up to this point, some educational software has been produced by the computer industry or the publishing companies. But most software has been constructed by computer hobbyists and classroom teachers. For a variety of reasons, most of these programs are not quality materials for general instructional use. Hobbyists enjoy exploring the unique features of the computer, often with little understanding of learning principles. Teachers often do not have mastery of advanced computer techniques. Both groups are giving us examples of first generation software for microcomputers. These examples will serve as materials on which future authors can improve.

The proliferation of educational software from numerous, nonstandard sources makes it hard to find materials for your purpose. One route to follow is the skimming of general software directories. (See Appendix B) Robert Purser's Magazine provides an excellent start. It lists software for several computers from any software group that submits information to Purser. Purser seeks to make the software known to users; he does little evaluation.

Radio Shack has started to publish a list of software

available for the TRS-80. At last count (1981), Sourcebook identified over 1000 of an estimated 8000 programs currently on the market for TRS-80 users in all fields. Other special machine sources are listed later.

Some educational catalogs are appearing. For example, Marck has recently initiated an educational catalog which makes software available for computers from selected commercial software houses. This catalog is well organized, but it does not contain much of the software available from individuals or small companies.

Clearinghouses of educational software are being formed with the major purpose of identifying quality materials. At the college level, Conduit has produced top quality materials and reviews. MicroSIFT, a similar clearinghouse for K-12 programs, is just beginning operations.

Identification and evaluation of selected software packages appear regularly in computer journals and magazines such as Creative Computing, Recreational Computing and The Computer Teacher. Subject area journals such as The Mathematics Teacher and The Journal of Computers in Science Teaching have software evaluation sections.

State and local users groups are valuable sources of information where such groups exist. The most notable group, the Minnesota Educational Computer Consortium (MECC), is a state-funded effort to support K-12 instructional computing. Some outstanding educational software came from MECC. Computer programs, often initiated by teachers in Minnesota

schools, are enhanced by a team of educational and computer specialists at MECC. These are then tested in schools throughout the state for further refinement. Write for the MECC newsletter and software directory. (Appendix B) Similar efforts are under way in other states but without state funding. The Puget Sound Program Library Exchange is an active group of this type.

The Michigan Association for Computer Users in Learning produces the MACUL Journal. Two recent issues (MACUL 1980, 1981) were devoted exclusively to evaluations of over 250 pieces of educational software. This ambitious project is one of the few published compendiums of educational software evaluations.

State or local subject area groups sometimes have active computer subgroups. The Wisconsin Mathematics Council has a major strand of instructional computer sessions at state meetings and includes articles in its publications. Computer subgroups are forming in science, media, business education, curriculum coordinators and other state organizations.

A multitude of local computer users' groups exist. Seek out the groups in your area. Many are machine, business or hobbyist oriented. There are growing numbers that have an educational focus. These are often centered in universities, regional educational agencies, computer stores or schools with energetic teachers. In users' groups, there is frequent clustering based on interests. This is your opportunity to study specific software in detail.

The previous paragraphs presented a variety of sources for the identification of educational software and opportunities for evaluation of that software. The next section discusses software that is currently available.

Educational Software Availability

There is an abundance of education-related software for the popular microcomputers such as Apple, PET and TRS-80, and there is a growing list of materials for the Atari. Unfortunately, very little of the software is well suited for sound instructional use.

At present there is a proliferation of drill and practice programs in arithmetic, reading, spelling, grammar and word lists for several topics. These range from routines that effectively blend graphics, timing and sound presentations to many which are boring, hard to follow and inaccurate. There are so many of these programs because they are relatively easy to write and require little content expertise by the programmer. A few more elaborate programs such as the Millikin Math Sequence have a computer-management component that records student progress.

Very little is available in tutorial software in the K-12 range. Some drill programs provide minimal remediation or clues. However, these should not be construed as instructional software.

There is a growing number of single-concept programs being developed. Many of these are adequate for use by the instructor as part of a class demonstration. Few of these

programs are of sufficient quality to stand alone in concept instruction.

There are numerous computer games enjoyed by students in the K-12 range. Often these are rejected by teachers without examination. Many of these games have educational value even if it is sometimes outside the narrowly defined set of school objectives. The better games place students in problem solving situations more challenging than those typically provided in the curriculum. Often they require students to develop complex strategies. Some games stress hand-eye coordination and reaction time for expert players. Out of these games, a few instructional objectives are even mastered. In the popular "pong" games, students quickly internalize the basic physics principle which states that the angle of incidence equals the angle of reflection. While such a principle cannot be stated by many pupils, they use it operationally.

The area of educational simulations is of major interest. Few teachers are skilled in this form of instruction, but computer software does provide this powerful teaching tool in business, industry, universities and the military (Dennis, 1979, No. 8E).

Some of the best secondary school simulation programs in the sciences and social sciences were developed on time-sharing computers in the Hungtington II Project (Digital Equipment Corporation). Some of these are being converted to microcomputer software by a variety of companies. The Oregon

Trail simulation is an excellent social studies example. The Three Mile Island simulation by Muse Software is a software package where one experiences the factors involved in a nuclear power station. Because of the complex interaction of content, learning principles and programming complexity, there is only a limited number of sound educational simulations on the market yet.

Selection of Software

Once a piece of software has been identified, the educator must start the evaluation process. Unfortunately, it is seldom possible to get software on a trial basis as is common with text materials. The problems of shipping a diskette that is to be returned and the ease of electronically copying diskettes make trial periods impractical. Consequently, initial information must be obtained from the distributor or from promotional materials (NCTM, 1981). Educational information requested should include: instructional objectives, grade levels or ability levels of intended audience, instructional materials available to support the program, and testing or validating results of the software in classroom settings. Technical information should include: the computer equipment required to run the software, the computer language(s) in which the software is written, and a listing of documentation available for purchase. Management information should include: cost of the software, cost of support materials, procedures and/or cost of receiving software for an in-depth evaluation, warranty and backup copy of software,

and cost of multiple copies for use in several district schools.

In addition to obtaining this information from the producer or distributor, it is worth the time to search for software reviews in appropriate journals or from other users. If the combined information suggests that the software may fit your goals, you are ready for a full-scale evaluation. If local computer stores do not stock the software, you may have to purchase it or send partial payment to review it.

When the software and accompanying materials arrive, you are ready to evaluate them for your instructional uses. This evaluation is best done in stages and with several individuals using the materials. The process should parallel constructive procedures used for textbook and materials evaluations. It is highly recommended that a checklist or evaluation form be used at each stage. It is important to record observations and ratings on items ranging from software execution on your computer to the educational merits of the program. Many checklists and forms exist (see Dennis, No. 5b, Appendix A; The Computing Teacher, 12-13). For general use, the author suggests a checklist similar to the NCTM (1981) example with its accompanying commentary. (Appendix C) This form has proven easy to use and productive in making purchase decisions by teachers.

A thorough checklist raises an important range of questions and encourages reviewer reactions on each point. Now let us consider a procedure for using it.

First, just load the software and run it. See if you can get into the learning environment without elaborate preparation. Does the software treat you humanely even when you are not sure what you are doing? This cursory overview can tell you how students might be handled when they will be working at the computer independent of a teacher. You might reject a product at this stage if it treats you poorly or throws you out of the program for making some honest mistakes.

Second, read the documentation and instructions carefully. Find out what the authors intended and learn how they viewed the program being used. Now go through a run of the software doing everything as it was designed to be done. For a student-centered program, enter the correct answers. On this pass, carefully evaluate the content and instructional sequence. Also, see if there is some life or enthusiasm such as graphical display, color or sound which will make the material motivating to your students. If the routine is boring and repetitive, students may quickly dread the assigned trip to the computer. Appropriate pacing is also important. If it is a demonstration program that does not enhance your content presentation, do you need it? Reflect thoughtfully on the software at this stage. You are seeing it at its best--when you could infer that learning is occurring.

Third, repeat stage two but do all the wrong things: respond incorrectly (i.e., give a word when a number is

expected or misspell the correct response), try to get out of the program and follow directions incorrectly. These are things that happen to a user who is having learning difficulties or is confused by the instructional sequence. While few existing software packages respond well to a full range of errors, we should search for those which do it best. If the student is going to be working independently at the computer, the ability to accept incorrect responses and keep the student on track is critical for a software package. If students are involved in a long, complex simulation and one input error destroys the results, they might refuse to start over. If one input error ruins a demonstration program, the teacher may find it less frustrating to omit the whole computer demonstration.

At this stage you are evaluating the software for its ability to keep the user on track within the program. For student users, you must evaluate the author's procedure for handling student errors.

Fourth, recruit a few students to run through the program. If it is a student-centered program, observe as each student runs the program. Do this with one or two students who have been through the content. Interview them after observing their performance. If the program is intended to teach a concept or skill, repeat the process with students who do not know the content. Observe and interview them. If the software is for a demonstration, give a mini-lesson to three students using it. Critique the lesson and computer features with them.

This stage seems time-consuming, but it can be very revealing in your attempt to evaluate a piece of software. Since most programs deal with only one or a few objectives, this student participation can give you valuable information on how the student sees the material. The interviews may often give you insights into how best to use the software within your course and into possible ways to modify the computer software if you keep it.

Stages one through four should be done by several potential users. The strengths, weaknesses and varied uses of a program as seen by several teachers may lead to creative use of a program for students.

In making a purchase decision, the reviewing staff might suggest possible changes in the software and associated materials that will enhance its use in their courses. It should be determined if such changes are possible. Is the program written in a language known well by teachers, students or volunteers in the district? If so, can they anticipate making those changes? With "no" answers, do not plan to make changes. With "yes" answers, the changes are possible, not guaranteed. Modifying another author's program is not a trivial assignment. Good documentation from the manufacturer is necessary for most revision plans.

Now you should be able to make a purchase decision. Good luck.

Developing Your Own Software

When you decide to develop your own software, you must

anticipate considerable time in planning, writing and refining the product. The critical phase is planning what the program should do in the learning process.

If your product centers on sequences of multiple-choice or true-false questions and responses, there are several author languages that allow you to enter text and choices for this format. Author languages such as PILOT, GENIS I and TUTOR are available for this task. These software packages have a preprogrammed structure and format in which you enter material for tutorial programs on computer-generated tests. In this limited form, you can quickly master the procedures.

If your project goes beyond the multiple-choice format, you must map out the expectations for computer use. A flow chart or other sequential description of the program steps is desirable. For most microcomputers, the programmer will use BASIC or possibly Pascal language. Mastery of one of these is a nontrivial task This is especially true if you are going to use the full capabilities of the computer. Competency with computer graphics (possibly including color graphics, animation, sound generation) and control of other devices such as a printer is very time-consuming to develop.

For a teacher with only introductory computer knowledge, the educational requirements can be spelled out. From that point, assistance from others may be necessary. Those who have developed expertise should be involved. Initial success can be achieved by a team effort of the teacher and one or more computer programmers. In school systems, advanced pro-

gramming students at the high school are potential experts. The author has had high school students who responded well in creating teacher-requested instructional programs.

In software construction, teachers learn the need to determine every action of the instructional program. Since a program may run independent of the teacher, most options must be anticipated and accounted for in the plan. This is typically not necessary in the classroom where the teacher can adapt to situations as they occur. The time taken to develop a total plan before writing computer instructions is well rewarded. The concept of structured programming is valuable in this context.

The complexity of developing good instructional software is clearly documented in the MECC (1980) publication, A Guide to Developing Instructional Software for the Apple II Microcomputer. Practical suggestions are given for program planning and construction. MECC stresses the separation of software author and computer programmer. Each person has an area of expertise. The key is the establishment of communication between the two.

Conclusion

Computer software is becoming available in increasing quantities. However, there is much to be desired in the quality of the materials on the market. The classroom teacher should thoroughly examine the documentation materials to learn the intended uses, content and supplementary resources required. Then the software should be used in several stages

to see if it is educationally sound from both teacher and student perspectives.

Since software is often designed to operate with little teacher intervention, the evaluation must determine if students can learn and interact meaningfully with it. The assignment of students to a computer lesson does not relinquish the teacher's responsibility for monitoring student progress.

References

CONDUIT (Clearinghouse-University), P.O. Box 388, Iowa City, IA 52244.

Courseware review and rating form. The Computing Teacher. December 1979-January 1980, 7 (3), 12-13.

Dennis, J. R. Computer Simulation and its Instructional uses. The Illinois series on educational applications of computers, No. 8E. Urbana, IL: Department of Secondary Education, University of Illinois, 1979.

Michigan Association for Computer Users in Learning. MACUL Journal. Spring 1980, III(2) and Winter 1981, IV (4). P.O. Box 1406 Washtenaw Intermediate School District, Ann Arbor, MI 48106.

MicroSIFT (Clearninghouse K-12), Northwest Regional Educational Laboratory, 710 S.W. Second Avenue, Portland, OR 97204.

Minnesota Educational Computing Consortium. A Guide to Developing Instructional Software for the Apple II Microcomputer. St. Paul, MN: MECC, 1980. 2520 Broadway Drive 55113

Milliken Math Sequence. Milliken Publishing, 1100 Research Blvd., St. Louis, MO 63132.

National Council of Teachers of Mathematics. Guidelines for Evaluating Computerized Instructional Materials. Reston, Va.: National Council of Teachers of Mathematics, 1981.

Rawitsch, D. Software Copying: Know the Law. Users: The MECC Instructional Computing Newsletter, November 1980, 7(2), 1980, p. 7.

Chapter 7

DEVELOPING MATERIALS FOR MICROCOMPUTERS

Michael L. Agin and Cynthia S. Simonsen
Michigan Technological University
Houghton, Michigan

Limitations of Available Programs

As you accumulate materials for your microcomputers you will find that there is much available, but many resources may not meet your classroom needs. Many authors have developed materials--programs, associated printed matter and other media--without much thought about pedagogical rules or instructional sequencing. They have often developed games, drill and practice, tutorial or problem solving packages without giving much attention to the needs of the learner or the teacher.

There are materials that have been well developed but still present problems for you, the teacher. Some of these are listed below.

1. Many programs have not been documented adequately so that you are not able to decipher the variables used or follow the instructional sequence.

2. Each program is the product of somebody elses thoughts and ideas. The author's view of the learning process may not agree with yours.

3. The materials may be too advanced or too general to suit your needs.

4. The sequencing of learning experiences may not be useful to you.

5. The time it would take to revise the program to meet your needs may be greater than the time you would need to develop your own program.

With all these concerns (obstacles) in mind, you should consider the possibility of developing your own program. You will be designing materials which reflect your personality and teaching styles, the materials you develop will be for your students, these materials will be completely adaptable to your needs, and they will be in the sequence you want. There are, however, some cautions which you should keep in mind as you pursue this alternative.

1. Before you start to develop an instructional package, be sure that a program which meets your needs is not already available. If a good program exists, use it!

2. Developing a good program is time consuming but worth the effort.

3. Initially develop one program at a time until you have gained confidence in your ability as a programmer.

4. Do a lot of paper work first. Set up your ideas and analyze them before you venture to put them into the microcomputer as a program.

5. Document your rationale and efforts through "remark" statements within your program and written explanations.

This chapter has been designed to introduce you to a logical and systematic approach to developing materials for microcomputers. We will discuss the development of modules and give you an example of one. Although this chapter is restricted to developing modules for microcomputers, a more extensive presentation of this subject is given by Dick and Carey (1978). They have presented a systems approach to developing modules for various teaching strategies including microcomputers.

Constructing Modules: A Systematic Approach

A module has been defined and described in various ways, but in general it is:

> An instructional package dealing with a single conceptual unit of subject matter. It is a self-contained entity that includes:
>
> 1. an integrated theme (main thread),
> 2. information about skills and knowledge to be acquired,
> 3. the use of various sets of material,
> 4. a means to measure student progress, and
> 5. a means to evaluate its effectiveness (Russell, 1974; Dick and Carey, 1978).

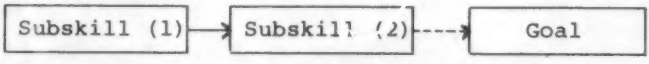
A module usually includes media in various combinations in order to extract the maximum benefit of each. In this chapter we will restrict ourselves to the microcomputer almost exclusively because we want to emphasize the logic of module development. But we recommend that you develop modules that provide student information from several sources.

The development of a module, especially for a microcomputer, requires a well organized logical approach. In its development there are points that need to be considered.

1. Decide on a main thread (or theme) that can be used for the module's framework. From this theme you select an instructional goal. Be sure to select an instructional goal for which you have the academic background to develop it adequately, the time to develop it properly, and students mature enough to handle it. In addition, you must be sure that the student can achieve the instructional goal in a reasonable length of time. In our opinion, modules of less than one hour and more than fifteen minutes are most desirable, but the theme should dictate the length.

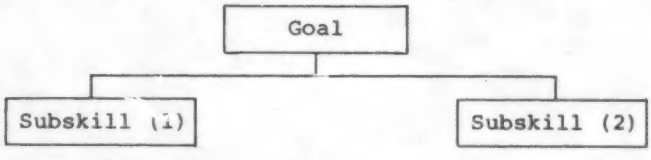
2. Once an instructional goal is established, identify the subskills and knowledge (subskill categories) that are necessary to achieve the goal. This process involves a top-to-bottom analysis (backward-stepping analysis). You start with the goal and work backwards to determine the subskills the students must possess to enable them to participate successfully. Continue backwards until you reach a desirable starting point. All subskills and knowledge before this point are entry level behaviors for your module.

3. While you are analyzing your instructional goal and dividing it into subcategories, decide how these subskill categories should be organized. The subcategories of some modules are sequential in nature. Students must successfully master one subskill before they can master the next.



In the above diagram, for example, the students need to master subskill (1) before they can be expected to master subskill (2) and so forth until they master the instructional goal.

Other modules have subcategories that include a series of branches in which each branch is independent of the other(s) but must be mastered before the instructional goal can be achieved.



In the example given above, the students must master a hierarchical series of skills before they can achieve the goal. But the mastery of subskill (1) is independent of the mastery of subskill (2). If a student cannot achieve the goal you must determine where their deficiency exists by testing for the mastery of subskill (1) and subskill (2) separately. Many other modules include subskills which are organized into a combination of sequential and hierarchical steps.

4. Once you have organized the steps of your module, you are ready to define each subskill in specific behavioral terms (performance objectives). Each subskill and the instructional goal should be defined as a simple performance objective which is used as a guide for developing the module into a microcomputer program. All activities of the module, whether they involve the microcomputer or other media, should be directed toward satisfying these objectives.

5. As you design your program, use the objectives to develop activities which will help the student achieve the instructional goal. This process is the reverse of goal analysis; you now work from the simpler to the more complicated subskills and ultimately to the instructional goal. While developing these teaching-learning episodes, remember the target population. You must constantly review your module development efforts to assure that you are working at an appropriate level of abstraction because the intellectual capabilities and prior knowledge of your students must be at a high enough level for them to participate successfully in the activities. A module appropriate for ninth graders may be fine for some but not all seventh graders.

6. The sequencing of activities in your program should follow approximately the organizational structure developed during goal analysis. It may be necessary to make two modules out of one because the size of the one module has become too large for the memory storage capacity of your microcomputer or the module takes too long for the student to complete. In some cases you may want to use subroutines for activities that are used several times in a module. In other instances, you may want to simplify your program by deleting extraneous material. (NOTE: There are ways of getting greater use of the microcomputer memory storage. Check with somebody who has a microcomputer like yours.)

7. While developing the module, investigate various methods of presenting material to students--text mode, graphic mode, printed narration, printed diagrams, etc.--so that you present material in the best way possible. In many instances, however, the text mode on a microcomputer will be sufficient. If it is,

use it. Do not use other media for "window dressing;" it is a waste of time and may complicate your program.

Whenever possible try to include practice activities for the students and to provide them with feedback. The feedback can be in the form of a simple statement or the microcomputer can be branched into a subroutine in which the students are given remedial exercises to perform when they make errors. You can branch your program to subroutines that provide enrichment activities to challenge students or to determine the upper limits of their knowledge.

8. The evaluation of student progress should be closely associated with the development of the program. Although many developers of modules recommend pre- and post-instructional tests, we feel that test items embedded within the module are more effective. These embedded test items give the student immediate feedback and you can provide your students with exercises for remediation when necessary. As an experienced teacher you should know where the students have learning problems and be able to design exercises within your programs to capitalize on this knowledge. When you ask questions and assign tasks that you know will cause difficulties, include embedded test items to determine mastery.

   An incorrect answer can be the triggering mechanism for sending a student to a subroutine in which they must do remedial work or at least review selected concepts. You can also include embedded test items that keep a student working on one concept until they have successfully answered a predetermined number of questions correctly. Embedded test items simulates closely the question and answer discussion sessions in class.

9. Module evaluation should start when you begin to develop your module. Keep looking for flaws in your logic and programming techniques. Keep asking: "Is there a simpler way?", "Will the student understand what I want?", and "Is there a better way to manipulate the data?"

   You will, of course, wish to pursue the evaluation of your module in a more formal way. Evaluation while development is in progress--

formative evaluation--can be accomplished in several ways.

a. A one-on-one trial of your module by colleagues is an appropriate first approach to checking your work. They can play the role of the student and perform the activities you have developed. While they are performing the activities they should look for logical, pedagogical and technical errors. Encourage them to critique your work honestly and be prepared to accept any constructive criticisms that they offer.

b. After revising and refining your module based upon peer feedback, try the module on a small group of students. Encourage these students to indicate any difficulties they experience in understanding the instructions of the module. This feedback as well as that from your colleagues will help you to revise and refine your work.

c. Test the module on a class to give it one final trial before general use. But remember that a module, like other teaching materials, must be revised and refined regularly Feedback from your students should be used for the continuous improvement of the module.

An Application

The procedures described in this chapter were used to develop a module for writing chemical formula equations. We decided that our instructional goal should be:

> "Given the names of the chemical reactants, the student will classify and write a balanced formula equation for the reaction if it occurs."

The analysis of this goal into subskills and knowledge revealed that students must be able to do several tasks and know several concepts to successfully achieve the instructional goal. These subcategories were organized into two

independent branches that included a combination of sequential and hierarchical steps. After the subskills were determined, a specific performance objective was written for each of them. The objectives for the subskills are listed in Table 7.1 and are keyed to the flowchart in Figure 7.1.

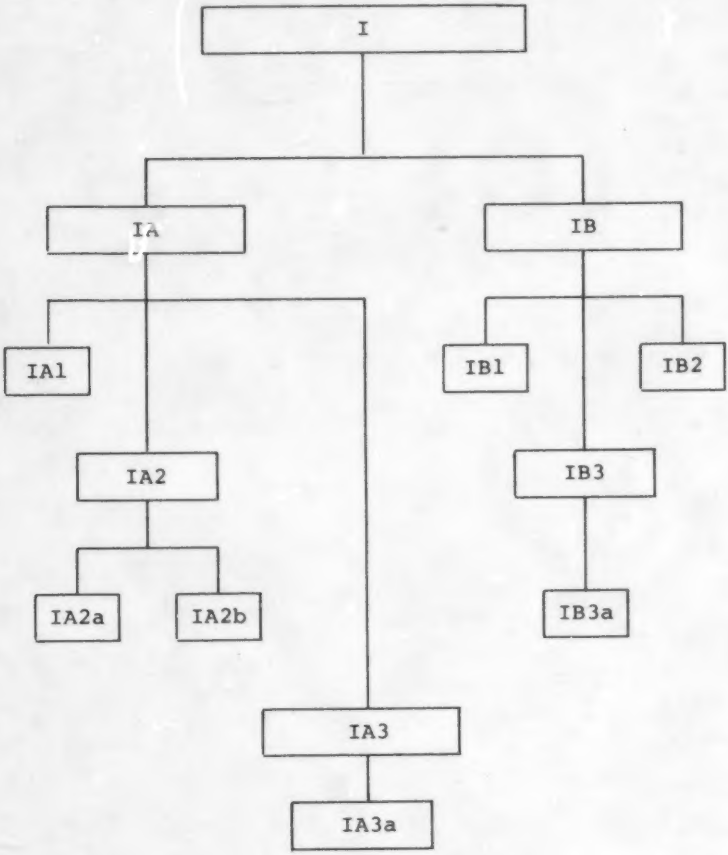


Figure 7.1. Flowchart for Writing Chemical Equations

Table 7.1

Writing Chemical Equations

I	Given the names of chemical reactants, the student will classify and write a balanced formula equation for the chemical reaction, if it occurs.
IA	Given a word equation, the student will write a balanced formula equation for the reaction.
IA1	Given the formulas of chemical reactants and products of a reaction, the student will balance the equation for that reaction.
IA2	Given its chemical name, the student will write the formula for the compound.
IA2a	Given the name of the element, ion, or radical, the student will write its symbol.
IA2b	Given the name of the element, ion, or radical, the student will write its oxidation number or valence.
IA3	Given the names of the chemical reactants, the student will write the names of the products that complete the word equation.
IA3A	Given the word equation of a chemical reaction, the student will identify the type of reaction that has occurred: (1) Composition (3) Single replacement (2) Decomposition (4) Double replacement
1B	Given the word (or formula equation) for a reaction, the student will indicate whether or not this reaction will occur (and go to completion).
IB1	Given the word equation for a chemical reaction, the student will indicate on the basis of the activity whether or not the reaction will occur.
IB2	Given the word equation for a chemical reaction, the student will indicate whether the reaction will go to completion by the formation of 1) H_2O, 2) Gas, or 3) Precipitate.
IB3	Given a chemical formula equation, the student will translate it into a word equation.
IB3a	Given their chemical symbols, the student will name the positive and negative ions or radicals of a compound.

Using these objectives a program was developed for the Apple II microcomputer that includes activities for practice, tasks to be mastered and embedded tests to measure the students' mastery of specific subskills. All the activities were directed toward assisting the students in achieving the instructional goal.

The program has been tested in a one-on-one situation and with small groups of teachers and students. It is being revised and refined according to suggestions that have been made for its improvement. For example, the program was designed to require a student to successfully complete a prescribed task ten times. However, we neglected to provide the students the option of quitting before finishing the program; some of them could have been trapped in the program for life. This aspect, as well as several others, is being modified to make the program more effective. In addition, information sheets need to be developed to supplement the microcomputer program.

We have not used the microcomputer program with classes but several teachers have made copies of it to be used in their microcomputer laboratories. No summative evaluation is available at this time.

In summary, we suggest that you:

1. Develop some of your own modules for microcomputers;
2. Start with a simple program and expand your efforts as you gain confidence;
3. Follow a logical procedure that includes embedded test items to measure student progress;

4. Seek and gracefully accept critiques of your module;

5. Revise and refine your module based upon these critiques; and

6. Be patient because it is easier to develop a module correctly than to have to revise it extensively.

References

Dick Walter and Lou Carey, The Systematic Design of Instruction, Glen Ellyn, Illinois: Scott, Foresman and Company (1978).

Russell, James D., Modular Instruction, Minneapolis, Minnesota: Burgess Publishing Company (1974).

Chapter 8

PROGRAMS FROM K-12 SCIENCE AND MATHEMATICS

Gary Marchionini and John Camp
Wayne State University
Detroit, Michigan

All computers require software. There are two general types of software: systems programs and applications programs. The purpose of this chapter is to illustrate how applications programs can be used to enhance the teaching and learning of science and mathematics.

Programs for Education

Because the tasks of teachers and students in teaching and learning are so varied, complex and individual, many types of application programs for education have emerged. These educational programs include the drill and practice, tutorial, simulations and games of CAI and the management programs for CMI. They can also include programs for special purposes such as graphing, data bases or statistical packages. A variety of application programs are discussed and examples are given for elementary, middle and high school science and mathematics.

Although most applications are purchased for use by teachers or students, many can be written by teachers or students who have some programming skills. This is especially true of some simple drill and practice, games and utility programs. Many educators believe that writing programs leads to superior understanding of the mathematics or science con-

cepts involved, and they assign programming tasks to develop these concepts. The development of concepts through programming activities need not entail the writing of a complete program. Programs that are complete except for key mathematical formulas for scientific facts can be assigned for student completion, or students can be asked to modify existing programs to generalize the results or complete a related task.

Further discussion and examples of educational applications of microcomputers can be found in commercial publications. Taylor (1980) presents a good collection of articles that extend the ideas introduced here and adds other applications. Doerr (1970) gives detailed examples of these and other educational applications. Professional periodicals regularly present new programs and ideas.

Examples of Applications Programs

The examples that follow were selected to illustrate various types of application programs for science and mathematics classrooms. The elementary programs are meant to be used by students with minimal computer experience. The middle school programs require more extensive user interaction, and they illustrate utility programs useful in mathematics and science. The high school programs and activities require extensive user interaction and, in the case of the mathematics examples, some BASIC programming skills. All programs were written in APPLESOFT BASIC and minor changes may be necessary for use on particular machines. A listing

of each program is followed by a sample run or runs of the program.

Example 1. Elementary School Science

Goals: Reinforce and develop vocabulary

Type of program: Drill and practice

Teaching notes: This program requires a screen output device. When run with a screen display, a science word appears for the requested amount of time (.1-5 seconds) and then disappears. Change the spelling words by replacing the words in line 400 with words of your choice. You may add other words by adding similar DATA statements between lines 400 and 500 and changing line 120 to FOR N=1 TO X where X is the number of words in your DATA lines.

]LIST (These are the statements which are typed into the computer)

```
90   REM              SPELLING DRILL
95   HOME
100  INPUT "WHAT IS YOUR NAME?";N$
110  INPUT "HOW MANY SECONDS? (.1 TO 5)";S
115  REM  N  IS THE NUMBER OF WORDS
120  FOR N = 1 TO 5
125  HOME
130  READ W$
132  VTAB 13:  HTAB 15
135  PRINT W$
137  REM  LINES 135-150 KEEP WORDS
138  REM  ON THE SCREEN REQUESTED TIME
140  FOR T = 1 TO S * 900:  NEXT T
150  HOME
155  VTAB 13:  HTAB 10
160  PRINT "SPELL THE WORD  ";
170  INPUT A$
180  IF A$ = W$ THEN 300
185  PRINT :  HTAB 10
190  PRINT "WRONG, TRY AGAIN";
200  INPUT A$
210  IF A$ = W$ THEN 300
212  PRINT
215  PRINT "THE WORD WAS";
220  PRINT W$
223  REM  4 SECOND PAUSE
```

```
225  FOR T = 1 TO 3600:  NEXT T
230  GOTO 370
300  GOSUB 500
340  PRINT N$
353  REM  3 SECOND TIME PAUSE
355  FOR T = 1 TO 2700:  NEXT T
360  C = C + 1
370  NEXT N
375  PRINT :  PRINT
380  PRINT N$: "YOU HAD";C;"CORRECT."
390  GOTO 1000
400  DATA  MASS,GRAVITY,FORCE,INERTIA,WEIGHT
500  REM  RANDOM PRAISE ROUTINE
510  X = INT (4 *  RND (1) + 1
520  PRINT : PRINT : HTAB 12
530  ON X GOTO 600,700,800,900
600  PRINT "GREAT!";
610  GOTO 950
700  PRINT "FANTASTIC";
710  GOTO 950
800  PRINT "WAY TO GO";
810  GOTO 950
900  PRINT "NICE JOB";
950  RETURN
1000 END
```

```
]RUN                                  (These lines appear on the mon-
WHAT IS YOUR NAME?  SUE               itor and on the printout if
HOW MANY SECONDS? (.1 to 5)1          there is one.  The underlined
                                      part is what the student enter-
                                      ed as a response to the ques-
                                      tion or directive.)
          MASS
     SPELL THE WORD    ?MASS

        GREAT! SUE
          GRAVITY
     SPELL THE WORD    ?GRAVITY

        FANTASTIC SUE
          FORCE
     SPELL THE WORD    ?FORCE

       WAY TO GO SUE
         INERTIA
     SPELL THE WORD    ?INERTA

     WRONG, TRY AGAIN ?ENERTA

THE WORD WAS INERTIA
          WEIGHT
     SPELL THE WORD    ?WATE
```

```
     WRONG, TRY AGAIN   ?WEIGHT

          NICE JOB SUE

SUE YOU HAD 4 CORRECT.
```

Example 2. Elementary School Mathematics

Goals: Reinforce concepts of greater than, less than, and between.

Type of program: Educational game

Teaching notes: Students should run the program several times. This may be done in small groups. Have students record the number of guesses for each game and discuss any strategies which students devise.

```
]LIST

20   PRINT "HELLO, WELCOME TO THE GUESS MY"
25   PRINT "NUMBER GAME.  I'M THINKING OF"
30   PRINT "A NUMBER BETWEEN 0 AND 100."
35   PRINT "TRY TO GUESS MY NUMBER IN LESS"
40   PRINT "THAN 20 TRIES "
45   REM  GENERATE RANDOM NUMBER 0-100
46   REM  AND INITIALIZE COUNTER FOR
47   REM  NUMBER OF GUESSES.
50   LET N = INT (100 *  RND (1)) + 1
60   LET C = 0
110  PRINT : PRINT
115  IF C = 20 THEN 500
120  PRINT "WHAT IS YOUR GUESS"
130  INPUT G
135  C = C + 1
140  IF G > N THEN 300
150  IF G < N THEN 400
155  PRINT : PRINT
160  PRINT "YOU GOT IT!"
170  PRINT "YOU MADE ":C:" GUESSES."
175  PRINT : PRINT
180  PRINT "DO YOU WANT TO TRY AGAIN?"
190  PRINT "TYPE YES OR NO"
200  INPUT A$
205  PRINT : PRINT
210  IF A$ = "YES" THEN 50
220  GOTO 1000
300  PRINT : PRINT
310  PRINT "TOO HIGH"
330  GOTO 110
```

```
400   PRINT : PRINT
410   PRINT "TOO LOW"
430   GOTO 110
500   PRINT "SORRY, YOU USED ALL YOUR GUESSES
510   PRINT "THE NUMBER WAS ":N
520   PRINT : PRINT
530   GOTO 180
1000  END

]RUN
HELLO, WELCOME TO THE GUESS MY NUMBER GAME.  I'M THINKING OF
A NUMBER BETWEEN 0 AND 100.  TRY TO GUESS MY NUMBER IN LESS
THAN 20 TRIES.

WHAT IS YOUR GUESS
?50

TOO HIGH

WHAT IS YOUR GUESS
?25

TOO HIGH

WHAT IS YOUR GUESS
?13

TOO HIGH

WHAT IS YOUR GUESS
?7

TOO HIGH

WHAT IS YOUR GUESS
?3

YOU GOT IT!
YOU MADE 5 GUESSES.

DO YOU WANT TO TRY AGAIN?
TYPE YES OR NO
?NO
```

Example 3. Middle School Science

Goals: Reinforce and extend concepts of mass and gravity by examining Newton's Law of Universal Gravitation; explore exponential relationships.

Type of program: Utility

Teaching notes: The program may be used to do the calculations necessary to complete a table of data that lists the masses of two objects, distance between them, and the gravitational force between them. Students use the completed table to answer questions such as the following:

1. How does doubling the mass of one of the objects effect the force between them? Tripling the mass?
2. How does doubling the distance between the objects effect the force between them? Tripling the distance?

The program prints the force in terms of G so that students may easily observe the effects of changing the mass and distance. To modify the program to calculate the force in newtons, replace line 180 by the following line:

```
180 PRINT "THE FORCE IS ";F*6.67E-10;" NEWTONS"
```

```
]LIST

10    REM  CALCULATES FORCE BETWEEN TWO
20    REM  OBJECTS USING NEWTON'S LAW OF
30    REM  UNIVERSAL GRAVITATION.
40    PRINT "YOU WILL BE ASKED TO INPUT THE MASS"
50    PRINT "OF TWO OBJECTS AND THE DISTANCE (IN METERS)"
60    PRINT "BETWEEN THEM.  THE FORCE BETWEEN "
70    PRINT "THE OBJECTS WILL BE GIVEN IN "
80    PRINT "G NEWTONS.  G IS THE UNIVERSAL "
90    PRINT "GRAVITATIONAL CONSTANT."
100   PRINT : PRINT
110   PRINT "WHAT IS THE MASS OF THE FIRST"
120   INPUT "OBJECT IN KG? ";M1
130   PRINT "WHAT IS THE MASS OF THE SECOND"
140   INPUT "OBJECT IN KG? ";M2
150   PRINT "WHAT IS THE DISTANCE BETWEEN"
160   INPUT "THE OBJECTS? ";D
170   F1 = (M1 * M2) / D ^ 2)
175   PRINT : PRINT
180   PRINT "THE FORCE IS ";F1;"G NEWTONS"
185   PRINT : PRINT
190   PRINT "DO YOU WANT ANOTHER?"
200   PRINT "TYPE Y FOR YES, N FOR NO."
210   INPUT A$
```

```
215  PRINT : PRINT
220  IF A$ = "Y" THEN 110
230  END

]RUN
YOU WILL BE ASKED TO INPUT THE MASS OF TWO OBJECTS AND THE
DISTANCE (IN METERS) BETWEEN THEM.  THE FORCE BETWEEN THE
OBJECTS WILL BE GIVEN IN G NEWTONS. G IS THE UNIVERSAL GRAVI-
TATIONAL CONSTANT.

WHAT IS THE MASS OF THE FIRST OBJECT IN KG? 10
WHAT IS THE MASS OF THE SECOND OBJECT IN KG? 10
WHAT IS THE DISTANCE BETWEEN THE OBJECTS? 4

THE FORCE IS 6.25G NEWTONS

DO YOU WANT ANOTHER?
TYPE Y FOR YES, N FOR NO.
?Y

WHAT IS THE MASS OF THE FIRST OBJECT IN KG? 20
WHAT IS THE MASS OF THE SECOND OBJECT IN KG? 10
WHAT IS THE DISTANCE BETWEEN THE OBJECTS? 4

THE FORCE IS 12.5G NEWTONS

DO YOU WANT ANOTHER?
TYPE Y FOR YES, N FOR NO.
?Y

WHAT IS THE MASS OF THE FIRST OBJECT IN KG? 30
WHAT IS THE MASS OF THE SECOND OBJECT IN KG? 10
WHAT IS THE DISTANCE BETWEEN THE OBJECTS? 4

THE FORCE IS 18.75G NEWTONS

DO YOU WANT ANOTHER?
TYPE Y FOR YES, N FOR NO.
?Y

WHAT IS THE MASS OF THE FIRST OBJECT IN KG? 10
WHAT IS THE MASS OF THE SECOND OBJECT IN KG? 10
WHAT IS THE DISTANCE BETWEEN THE OBJECTS? 8

THE FORCE IS 1.5625G NEWTONS

DO YOU WANT ANOTHER?
TYPE Y FOR YES, N FOR NO.
?Y

WHAT IS THE MASS OF THE FIRST OBJECT IN KG? 10
WHAT IS THE MASS OF THE SECOND OBJECT IN KG? 10
WHAT IS THE DISTANCE BETWEEN THE OBJECTS? 12
```

```
THE FORCE IS .694444444G NEWTONS

DO YOU WANT ANOTHER?
TYPE Y FOR YES, N FOR NO.
?N
```

Example 4. Middle School Mathematics

Goals: Extend data generated in probability experiment

Type of program: Simulation/utility

Teaching notes: Students should roll two dice and collect data for analysis before running the program. Distributions graphs and expected value can be discussed. Based on their experiences from rolling the dice and observing results, ask students to predict frequencies for 10,000 rolls. Use the programs to simulate rolling the dice 10,000 times. The program takes about five and a half minutes to execute. Repeat the procedure and compare results. How do actual outcomes compare to the expected outcomes? To each other?

```
]LIST

50   DIM A(12)
100  INPUT "NUMBER OF ROLLS?"; ROLL
120  FOR N = 1 TO ROLL
140  D1 = INT (6 * RND (1)) + 1
160  D2 = INT (6 * RND (1)) + 1
180  SUM = D1 + D2
200  A(SUM) = A(SUM) + 1
220  NEXT N
230  PRINT : PRINT
240  PRINT "FOR ":ROLL:" ROLLS THERE WERE:"
250  PRINT : PRINT
260  PRINT "TWOS", "THREES", "FOURS"
280  PRINT A(2),A(3),A(4)
290  PRINT
300  PRINT "FIVES","SIXES","SEVENS"
320  PRINT A(5),A(6),A(7)
330  PRINT
340  PRINT "EIGHTS","NINES","TENS"
360  PRINT A(8),A(9),A(10)
370  PRINT
380  PRINT "ELEVENS","TWELVES"
```

```
400  PRINT A(11),A(12)
500  END

]RUN
NUMBER OF ROLLS? 10000

FOR 10000 ROLLS THERE WERE:

TWOS          THREES        FOURS         FIVES
344           423           991           1186

SIXES         SEVENS        EIGHTS        NINES
1414          1505          1380          1123

TENS          ELEVENS       TWELVES
914           433           287


]RUN
NUMBER OF ROLLS? 10000

FOR 10000 ROLLS THERE WERE:

TWOS          THREES        FOURS         FIVES
396           297           1188          1287

SIXES         SEVENS        EIGHTS        NINES
1485          1386          1287          1090

TENS          ELEVENS       TWELVES
990           297           297
```

Example 5. High School Science

Goals: Various concepts in chemistry, biology, or physics can be developed via simulations.

Type of program: Simulation

Teaching notes: Simulations that demand extensive user interaction and provide a realistic range of options require a great amount of programming code. Therefore, no listings or runs are given here. However, some programs which are commercially available are discussed.

A number of simulation programs and related course materials were produced by the NSF supported Huntington Pro-

ject. (See Appendix B.) Many of the programs are now commercially available through software vendors. Some publishers have added graphics enhancements and provide written documentation and suggested classroom uses. The Minnesota Educational Computing Consortium (MECC) also has produced a number of simulations for science classroom use. Some examples of each project's simulations follow.

1. RATS is a program that requires the user to enter the garbage level, type of rat poison and quantity of poison to control a rat population in a city or apartment building. (Huntington)

2. BUFFALO is a program that allows the user to manage a herd of buffalo by manipulating reproductive rates and initial herd perameters. (Huntington)

3. MALARIA requires the user to manage a budget to contain a malaria epidemic. Money can be allocated to field hospitals, preventive drugs, treatment drugs and pesticides. (Huntington)

4. DIFFUSION allows the user to examine the diffusion rates of any number of gases in any single gas. (MECC)

5. COLLIDE allows the user to graphically view the collision of two bodies. The user controls the mass and velocity of the two bodies and specifies the elasticity of the matter composing the bodies. (MECC)

Example 6. High School Mathematics

Goals: Reinforce concepts of perimeter and area, introduce maximizing and minimizing related quantities.

Type: Problem solving/programming

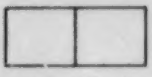
Teaching notes: Discuss getting the most area from a given amount of fencing. For example, 400 meters of fencing will enclose what maximum area? Collect data and graph the area versus length (or width). The resulting parabola can lead to a generalization about the optimal rectangle for maximizing area. Try some other cases by asking students to write and run short programs to calculate values for any given perimeter. The writing of the programs requires the student to use the relationships among the area, length, width and perimeter. If students are unable to write programs, have them supply the formula used in line 40 of the first program below. Two example programs and the resulting output are given. Note that the second program is more generalized than the first in that it allows more user control over the parameters. Programs of this type are usually run several times in order to complete the problem. Keeping the perimeter constant and varying the interval of increment allows the user to focus on the maximum area.

]LIST (Program 1 on Area)

```
10    REM  MAXIMUM RECTANGULAR AREA
20    INPUT "ENTER PERIMETER ";P
30    FOR L = 1 TO P / 2 STEP 5
40    W = (P - 2 * L) / 2
50    A = L * W
60    PRINT A,
70    NEXT L
80    END

]RUN
ENTER PERIMETER 200
99            564           979           1344          1659
1924          2139          2304          2419          2484
2499          2464          2379          2244          2059
1824          1539          1204          819           384
```

```
]LIST (Program 2 on Area)

10    REM  MAXIMUM RECTANGULAR AREA WITH OPTIONS
20    INPUT "PERIMETER";P
30    INPUT "MINIMUM LENGTH (BEGINNING)";B
40    INPUT "MAXIMUM LENGTH (ENDING)";E
50    INPUT "INCREMENT";I
60    FOR L = B TO E STEP I
70    W = (P - 2 * L) / 2
80    A = L * W
90    PRINT A,
100   NEXT L
110   END
120   REM  GOOD EXAMPLE FOR GIGO (GARBAGE IN GARBAGE OUT).  IF
      PARAMETERS ARE NOT CAREFULLY SELECTED, NEGATIVE AREAS
      RESULT!

]RUN
PERIMETER 200
MINIMUM LENGTH (BEGINNING) 10
MAXIMUM LENGTH (ENDING) 90
INCREMENT 5
900          1275          1600          1875          2100
2275         2400          2475          2500          2475
2400         2275          2100          1875          1600
1275         900
```

After students run the programs, ask them to modify their programs to solve the problem given that the fenced area is to be divided into two corrals as in this illustration.



The key change in the program is in line 70 where a new formula for determining the width is needed. The following program also adds some output modifications. Run the program for several different perimeters and record the optimal length and width for each. Is the ratio of width to length constant for all perimeters?

```
]LIST (Program 3 on Area)

10    REM  MAXIMUM RECTANGULAR AREA WITH OPTIONS
15    REM  TWO CORRALS
20    INPUT "PERIMETER";P
```

```
30    INPUT "MINIMUM LENGTH (BEGINNING)";B
40    INPUT "MAXIMUM LENGTH (ENDING)";E
50    INPUT "INCREMENT";I
53    PRINT "FOR A PERIMETER OF ";P
54    PRINT
56    PRINT "LENGTH","WIDTH","TOTAL AREA"
57    PRINT
60    FOR L = B TO E STEP I
70    W = (P - 2 * L) / 3
80    A = L * W
90    PRINT L,W,A
100   NEXT L
110   END

]RUN
PERIMETER 200
MINIMUM LENGTH (BEGINNING) 10
MAXIMUM LENGTH (ENDING) 90
INCREMENT 5
FOR A PERIMETER OF 200

LENGTH                    WIDTH                    TOTAL AREA

10                        60                       600
15                        56.6666667               850
20                        53.3333333               1066.66667
25                        50                       1250
30                        46.6666667               1400
35                        43.3333333               1516.66667
40                        40                       1600
45                        36.6666667               1650
50                        33.3333333               1666.66667
55                        30                       1650
60                        26.6666667               1600
65                        23.3333333               1516.66667
70                        20                       1400
75                        16.6666667               1250
80                        13.3333333               1066.66667
85                        10                       850
90                        6.66666667               600

]RUN
PERIMETER 200
MINIMUM LENGTH (BEGINNING) 45
MAXIMUM LENGTH (ENDING) 55
INCREMENT 1
FOR A PERIMETER OF 200

LENGTH                    WIDTH                    TOTAL AREA

45                        36.6666667               1650
46                        36                       1656
47                        35.3333333               1660.66667
48                        34.6666667               1664
```

LENGTH	WIDTH	TOTAL AREA
49	34	1666
50	33.3333333	1666.66667
51	32.6666667	1666
52	32	1664
53	31.3333334	1660.66667
54	30.6666667	1656
55	30	1650

The problem can be further extended by increasing the number of corrals required. Allow the user to input a value (e.g., 5) for the number of corrals and change the divisor in line 70 to 5. Run the program for several perimeters and several numbers of corrals. Is the optimal case ratio of width to length constant for all perimeters? Is there a pattern to the various ratios for different corral arrangements?

Concluding Comments

The programs above were meant to illustrate a variety of types of computer programs. It is important to note that a particular program could be used in a number of instructional settings. Students at various grade levels could experience a program like the game program of Example 2 depending on whether they were using, writing, modifying or predicting the outcomes of the program. Simple programs can often serve as bases for elegant and powerful programs or concepts.

A key to instructional computing lies in the creative use of programs. Teachers must be alert to opportunities to apply instructional computer programs at appropriate times. Computers can be powerful tools in the teaching/learning

process. As with any tool, the outcomes of applications depend on the appropriate use of the tool for a given situation.

References

Doerr, Christine. Microcomputers and the 3 r's. Rochelle Park N.J.: Hayden Book Company, 1979.

Taylor, Robert (ed.). Computers in the School: Tutor, Tool, Tutee. New York: Teachers College Press 1980

APPENDIX A

TERMS AND CLARIFICATIONS

(1) ASCII: The acronym for American Standard Code for Information Exchange, one popular system for representing symbols with a sequence of binary numbers.

(2) BASIC: The acryonym for Beginners All-purpose Symbolic Instructional Code, a programming language developed by John Kemeny and Thomas Kurtz at Dartmouth College in the mid 1960's.

(3) Bit: A single binary digit (0 or 1).

(4) A byte typically reflects a single character such as a letter, number or other symbol. A byte associated with 8-bit microcomputers contains 8 binary digits.

(5) Diskette: A flexible plastic disk (resembling a phonograph record) which is used to store programs for microcomputers. Diameter is 5¼ inches. They are also referred to as "disks" or "floppy disks."

(6) The unit K actually represents 1024 bytes and not 1000 bytes.

(7) Microcomputer: As employed in this monograph, an FCC Class B computer. It employs a microprocessor and has memory. Most microcomputers also include a keyboard. Also commonly known as a personal computer or desktop computer. Handheld computers are not the type under consideration in this monograph.

(8) Microcomputer System: The combination of a microcomputer together with other peripherals such as visual display device, cassette deck, floppy disk, hard disk, game paddles, modem, graphics tablet, light pens and printers.

(9) Pascal: A programming language named after Blaise Pascal, a 17th century French mathematician who designed one of the first mechanical calculators in the world.

(10) Programable read only memories (PROM's), ultra violet erasable programable read only memories (EPROM's) and electrically erasable programable read only memories (EEPROM's) are not discussed in this monograph.

APPENDIX B

SOFTWARE INFORMATION

Apple Education News, P.O. Box 20485, San Jose, CA 95160. (A newsletter of Apple Computer, Inc.)

Apple Educators' Newsletter, 9525 Lucerne Street, Ventura, CA 93004. ($15/year, minimum of 6 issues.)

The Book of Apple Computer Software 1981. The Book Company., 16720 Hawthorne Blvd. Lawndale, CA 90260

Classroom Computer News, Box 266, Cambridge, MA 02138. (1 year, 6 issues, $9.00.) Newspaper format. Includes information on what is new in educational computer applications, products, research, grants and government programs.

COMPUTE, P.O. Box 5119, Greensboro, NC 27403. (1 year, 6 issues, $12.00.) Each issue contains both hardware and software information for machines using the 6502 chip, such as Apple, PET, Atari, and Ohio Scientific.

The Computer Teacher, Computing Center, Eastern Oregon State College, LaGrande, OR 97850. (1 year, 7 issues, $10.00.) The journal of computing for elementary and secondary education, and the impact of computers on curriculum.

Creative Computing, P.O. Box 789-M, Morristown, NJ 07960. (1 year, 12 issues, $15.00.) Monthly evaluations and profiles of software and hardware. Always includes sections on applications, games, problems and other special interests. Regular columns speak directly to users of the popular micro's (Apple, TRS-80, PET). Many program listings of high programming quality are included. One of the earliest, and still best, computer publications.

Huntington II Project Simulations, Digital Equipment Corporation, Software Distribution Center, Maynard, MA 01754.

Instant Software, Peterborough, NH 03458. Wide range of computer software including instructional materials, games, simulations. Software is guaranteed. Write for information.

The Journal of Computers in Science Teaching, Gary H. Marks, editor, P.O. Box 4455, Austin, TX 78765.

K-12 Micro Media, P O. Box 17, Valley Cottage, NY 10989.

Marck, 280 Linden Ave., Branford, CT 06082. (Apple, Atari, PET, TRS-80.) Write for catalog and further information.

MECC Instructional Newsletter, 2520 Broadway Drive, St. Paul, MN 55113

Micros in Mathematics Quarterly. Iowa Microcomputer Curriculum Project, Price Laboratory School, University of Northern Iowa, Cedar Rapids IA 50613

Puget Sound Program Library Exchange 517 Eleventh Ave., East, Olympia, WA 98102. A major source of computer software developed by individuals.

Robert Purser's Magazine, P O. Box 466, El Dorado, CO 95623. (Apple, Atari, TRS-80.) ($4.00 per issue, quarterly.) One of the best known software directories in the country. One issue had over 700 manufacturers of software for Apple, Atari and TRS-80. A free-lance magazine that tries to keep up on all that is becoming available.

Recreational Computing, P.O. Box E, Menlo Park, CA 94025. (1 year, 6 issues, $12.00.) This is recommended as a major source of new ideas Good articles by some really creative thinkers can provoke some excellent ideas for simulations. These can lead to some large-scale problem solving situations. There is a programming problems section of particular interest to teachers of computing.

Softside, Softside Publications, 6 South Street, Milford, NH 03055. (1 year, 12 issues, $24.00.) A variety of software articles, program listings, and applications packages for all machines.

TRS-80 Applications Software Sourcebook. Extensive list of software available for TRS-80 computers in all fields. Check your local Radio Shack computer store.

Three Mile Island. Muse Software, 330 N. Charles St., Baltimore, MD 21234.

NCTM SOFTWARE EVALUATION CHECKLIST

PROGRAM NAME:_____ SOURCE:_____ COST:_____

SUBJECT AREA:_____ REVIEWER'S NAME:_____ DATE:_____

1. INSTRUCTIONAL RANGE

_____ grade level(s)

_____ ability level(s)

2. INSTRUCTIONAL GROUPING FOR PROGRAM USE

_____ individual
_____ small group (size:_____)
_____ large group (size:_____)

3. EXECUTION TIME

_____ minutes (estimated) for average use

4. PROGRAM USE(S)

_____ drill or practice
_____ tutorial
_____ simulation
_____ instructional gaming
_____ problem solving
_____ informational
_____ other (_____)

5. USER ORIENTATION: INSTRUCTOR'S POINT OF VIEW

low				high	
*	*	*	*	*	flexibility
*	*	*	*	*	freedom from need to intervene or assist

6. USER ORIENTATION STUDENT'S POINT OF VIEW

low				high	
*	*	*	*	*	quality of directions (clarity)
*	*	*	*	*	quality of output (content and tone)
*	*	*	*	*	quality of screen formatting
*	*	*	*	*	freedom from need for external information
*	*	*	*	*	simplicity of user input

7. CONTENT

low				high	
*	*	*	*	*	instructional focus
*	*	*	*	*	instructional significance
*	*	*	*	*	soundness or validity
*	*	*	*	*	compatibility with other materials used

8. MOTIVATION AND INSTRUCTIONAL STYLE

passive				active	
*	*	*	*	*	type of student involvement

low				high	
*	*	*	*	*	degree of student control

none	poor			good	
*	*	*	*	*	use of game format
*	*	*	*	*	use of still graphics
*	*	*	*	*	use of animation
*	*	*	*	*	use of color
*	*	*	*	*	use of voice input and output
*	*	*	*	*	use of nonvoice audio
*	*	*	*	*	use of light pen
*	*	*	*	*	use of ancillary materials
*	*	*	*	*	use of _____

9. SOCIAL CHARACTERISTICS

present and negative	not present	present and positive	
_____	_____	_____	competition
_____	_____	_____	cooperation
_____	_____	_____	humanizing of computer
_____	_____	_____	moral issues or value judgments
_____	_____	_____	summary of student performance

1. The grade levels and ability levels for a particular program are primarily determined by the concepts involved. Other important factors are reading level, prerequisite skills, degree of student control, and intended instructional use. It is possible for a program to be flexible enough to be used across a wide range of grade levels and ability levels.

2. Some programs are designed for use by individuals. Others have been or can be modified for participation by two or three persons at a time. Simulations or demonstrations often pose opportunities for large-group interaction. A given program may be used in more than one grouping, depending on the instructor.

3. The time required for the use of a program will vary considerably. Include loading time for cassettes. A time range is the appropriate response here.

4. Instructional programs can be categorized according to their uses. Some programs may have more than one use. thus falling into more than one of the following categories:

Drill or practice: Assumes that the concept or skill has been taught previously.

Tutorial: Directs the full cycle of the instructional process; a dialogue between the student and the computer.

Simulation: Models selected, alterable aspects of an environment.

Instructional gaming: Involves random events and the pursuit of a winning strategy.

Problem solving: Uses general algorithms common to one or more problems.

Informational: Generates information (data).

5. These are factors relevant to the actual use of the program from the point of view of an instructor.

Flexibility: A program may allow the user or the instructor to adjust the program to different ability levels, degrees of difficulty, or concepts.

Intervention or assistance: A rating of "low" means considerable teacher intervention or assistance is required.

6. These are factors relevant to the actual use of the program from the point of view of a student.

Directions: The directions should be complete, readable, under the user's control (e.g., should not scroll off the screen until understood), and use appropriate examples.

Output: Program responses should be readable, understandable, and complete. If in response to student input, the output should be of an acceptable tone and consistent with the input request.

Screen formatting: The formats during a program run should not be distracting or cluttered. Labels and symbols should be meaningful within the given context.

External information: A program may require the user to have access to information other than that provided within it. This may include prerequisite content knowledge or knowledge of conventions used by the program designer as well as maps, books, models, and so on.

System errors: System errors result in the involuntary termination of the program.

Input: A program should ensure that a user knows when and in what form input is needed. It should avoid using characters with special meanings, restrict input locations to particular screen areas, and require minimal typing.

7. These are matters relevant to the subject-matter content of the program.

Focus: The program topic should be clearly defined and of a scope that permits thorough treatment.

Significance: The instructional objectives of the program must be viewed as important by the instructor. Also, the program should represent a valid use of the computer's capabilities while improving the instructional process.

Soundness or validity: The concepts and terms employed should be correct, clear, and precise. Other important factors are the rate of presentation, degree of difficulty, and internal consistency.

Compatibility: The content, terminology, teaching style, and educational philosophy of the program should be consistent with those generally encountered by the student.

9. Competition, cooperation, and values are concerns that may be a function of the way a program expresses them. (War gaming and the "hangman" format are sample issues.) Also, the "humanizing" of the computer may serve for motivation or to reduce anxiety, but it also may become tedious, misleading, and counterproductive.

The summary of student performance can be dichotomous (win or lose), statistical (time expended or percent of items correct), or subjective (as in the evaluation of a simulation). It may be for student, teacher, or both.

* * *

Reprinted with written permission of the
National Council of Teachers of Mathematics